JOURNAL OF COMPUTER AND INFORMATION SCIENCES

SAKARYA UNIVERSITY

e-ISSN 2636-8129 | VOLUME 4 | ISSUE 2 | AUGUST 2021

Analysis of C-shaped Compact Microstrip Antennas Using Deep Neural Networks Optimized by Manta Ray Foraging Optimization with Lévy-Flight Mechanism

Deep Gated Recurrent Unit for Smartphone-Based Image Captioning

Experimental Analysis of Energy Efficient and QoS Aware Objective Functions for RPL Algorithm in IoT Networks

Blockchain-Based Secure Credit Card Storage System for E-Commerce

www.saucis.sakarya.edu.tr

A Hybrid Prognostic Approach Based on Deep Learning for the Degradation Prediction of Machinery

Diagnosing Hematological Disorders Using Deep Learning Method

Base Station Power Optimization for Green Networks Using Reinforcement Learning

A Curve Fitting Modelling Approach to Forecast Long-Term Electrical Energy Consumption: Case Study of Turkey

Sakarya University Journal of Computer and Information Sciences
Volume: 4 – Issue No: 2 (August 2021)
http://saucis.sakarya.edu.tr/issue/64707

Editor in Chief

Associate Editors

Editorial Assistants - Secretary

Editorial Board

SAUCIS

SAUCIS

Sakarya University Journal of Computer and Information Sciences
Volume: 4 – Issue No: 2 (August 2021)
http://saucis.sakarya.edu.tr/issue/64707

Contents

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 4, NO. 2, AUGUST 2021

Research Article

Analysis of C-shaped Compact Microstrip Antennas Using Deep Neural Networks Optimized by Manta Ray Foraging Optimization with Lévy-Flight Mechanism

Mustafa Berkan Biçer[1]
[1]Department of Electrical and Electronics Engineering, Izmir Bakircay University, Izmir, Turkey;
mustafa.bicer@bakircay.edu.tr; +90 (232) 493 00 00 / 11237

Received 25 March 2021; Revised 27 April 2021; Accepted 29 April 2021; Published online 31 August 2021

Abstract

In recent years, microstrip antennas have become a popular research subject with the increasing use of mobile technologies. With the development of neural networks, the design and analysis of microstrip antennas are carried out quickly with high accuracy. However, optimizing the weight matrices and bias vectors of deep neural learning models is an important challenge for engineering problems. This study presents a deep neural network-based (DNN-based) neural model to estimate the gain and scattering parameter (S_{11}) of C-shaped compact microstrip antennas (CCMAs). For this purpose, the S_{11} and gain values of 324 CCMAs with different physical and electrical properties were obtained using full-wave electromagnetic simulation software based on the finite integration technique (FIT). The data related to 324 CCMAs were used for the training and testing process. The improved manta ray foraging optimization (MRFO) algorithm based on the Lévy-flight (LF) mechanism was used to optimize the connection weights matrices and bias vectors. The MRFO-optimized model has estimation success for training and testing data as 0.925 and 0.922, in terms of R^2 score, respectively. The estimated resonant frequencies using the trained model are compared with the studies in the literature, and an average percentage error (APE) of 0.933% is obtained.

Keywords: C-shaped microstrip antenna, deep neural networks, manta ray foraging optimization, lévy flight technique, S-parameter estimation, gain estimation

1. Introduction

With the rapid development of technology, small-sized wireless and portable devices such as navigation devices, headphones, cameras, and mobile phones and have found a wider area of use in daily life. Microstrip antennas (MA) [1, 2] are essential components of the systems such as wireless communication, spacecraft, aircraft, radars, satellite communication, and guided missiles. MAs have superior advantages, including ease in production, small in size, integrated use with electronic circuit boards, and solid-state devices [2–6]. Recent advances in mobile communication technology have increased the need for MAs operating at lower frequency bands. Due to its nature, the reduction in the physical dimensions of the antenna causes an increase in the resonant frequency of the antenna. However, the resonant frequency of conventional-shaped MAs such as circular, triangular, and rectangular can be reduced by modifying the structures [2, 5]. Compact MAs (CMAs) are formed by modifying the geometries of conventional-shaped MAs [2, 7]. Although theoretical methods, including the transmission-line model (TLM) [8] and cavity model [9], are used to analyze conventional-shaped MAs, these methods have various difficulties for CMAs due to their complex structure and non-linear behavior. Therefore, the analysis of CMAs is generally carried out using full-wave electromagnetic solvers based on numerical methods such as finite-difference time-domain (FDTD) [10, 11], finite element method (FEM) [12], finite integration technique (FIT) [13, 14], and method of moments (MoM) [15, 16]. In recent years, researchers have vigorously challenged the accurate calculation of the resonant frequency of CMAs. While most studies focus on designing and optimizing CMA structures [3, 17–20] and obtaining mathematical models for determining the resonant frequencies of the antennas [5, 7, 21–24], there are also studies on computing the resonant frequencies or other characteristic parameters using neural network models [25–34]. In most recent studies, resonant frequency estimation was made in two ways: the closed-form expression approach and the neural model approach. Akdagli et al. [7] have

To Cite This Article: M.B. Bicer, “Analysis of C-shaped Compact Microstrip Antennas using Deep Neural Networks optimized by Manta Ray Foraging Optimization with Lévy-Flight Mechanism,” Sakarya University Journal of Computer and Information Sciences, vol. 4, no. 2, pp. 166–180, 2021. doi: 10.35377/saucis.04.02.903208

proposed a closed-form resonant length expression to compute the dominant operating frequency of C-shaped compact microstrip antennas (CCMAs). To optimize the mathematical model, the authors have designed and simulated 144 CCMAs with various physical and electrical properties and utilized the artificial bee colony (ABC) algorithm [35–37]. In [23], mathematical expressions for resonant lengths of C-shaped and H-shaped CMAs are presented. Toktas et al. made simulations for certain parameter ranges, and the unknown variables of the expressions were determined employing the ABC algorithm [35–37]. Biswas and Dam [24] have introduced a closed-form mathematical model to calculate the probe reactance. In their study [24], an equilateral triangular patch antenna was fabricated, and the calculated resonant frequencies were compared with the measured and simulated resonant frequencies. A further equilateral triangle-shaped compact microstrip antenna study has been carried out by Kapusuz et al. [31]. In [31], the authors presented an artificial neural network (ANN) to determine the resonant frequency of equilateral triangular patch antennas for different TM_{mnp} modes. Can et al. have proposed a multilayer perceptron (MLP) neural network (NN) model to calculate the operating frequency of coaxial-fed pin-loaded rectangular microstrip patch antennas [32]. The proposed MLPNN model [32] uses the physical and electrical properties of the antenna as input and estimates the upper and lower operating frequency of the antenna. Abbassi et al. [38] have designed a CMA operating at a frequency of 2.4 GHz and used a NN model to predict the S_{11} and gain parameters. Neebha and Nesasudha [34] have designed a CMA for C-band applications using an ANN model. Four antenna-related parameters were used as the network input, and two physical parameters were estimated [34]. Singh et al. [33] studied on a coaxial-fed E-shaped CMA (ECMA) operating at the 2.4 GHz frequency. In [33], to analyze the performances of the neural network models, the x- and y- coordinates of the coaxial feed were used as input, and resonant frequency, S_{11} value, VSWR value, and input impedance parameters were chosen as output. Sami et al. have proposed an ANN model to estimate the resonant frequency and quality factor of a circular microstrip antenna in the TM_{11} mode [25]. The authors have used isotropic substrate and uniaxially anisotropic substrate in MA design [25]. In another study, a neural model for the synthesis and analysis of rectangular MAs using uniaxially anisotropic substrates was developed by Barkat et al. [26]. The patch dimensions were used for synthesis, and the resonant frequency and bandwidth parameters were used for analysis [26]. In [30], Kaur and Sivia have proposed a miniaturized hybrid fractal antenna based on Giuseppe Peano and Cantor set fractals for biomedical systems. While the firefly algorithm (FA) [36, 39] was used to optimize the feeding point of the antenna [30], ANN was employed in the calculation of the two resonant frequencies. In a recently published study [29], a deep neural network-based (DNN-based) model consisting of five layers, three of which are hidden layers, has been proposed by Ustun et al. to calculate the resonant frequency of ECMAs. The authors simulated 144 ECMAs and trained the neural model using the simulated data [29]. As mentioned above, several studies on the calculation or estimation of the resonance frequency of MAs have been done in the literature, but there is still insufficient study for the S_{11} and gain estimation. Studies on neural networks in the literature generally focused on estimating the resonance frequencies of antennas and their gains at this frequency. However, in this study, the estimation of S_{11} and gain values of the antenna not at a single frequency but in a certain frequency range was studied. S_{11} is a critical parameter that indicates whether the antenna carries the desired radiation performance in a frequency range and whether impedance matching is achieved in this range. The gain is also an important parameter as it takes the losses into account, representing the transmitted power in any direction with regard to an isotropic antenna. Hence, this study is focused on obtaining the S_{11} and gain curves of CCMAs designed. It is seen from the literature that the computer-aided design (CAD) tools and machine learning-based models are of great importance in guiding the antenna design process.

In this study, a DNN-based neural model is proposed to estimate the S_{11} and gain parameter of CCMAs for a frequency range. CCMAs with a specific value range of physical and electrical parameters were designed and simulated using a full-wave electromagnetic solver based on FIT. The obtained S_{11} and the gain values for the desired frequency range were used to build a data set. The neural model was trained by utilizing the improved manta ray foraging optimization (MRFO) algorithm [40] based on the Lévy-flight (LF) mechanism [36, 41]. The estimated values were compared with the studies in the literature.

This paper is divided into six sections. In the first section, the definition of the problem is given by doing a literature review. The covers the antenna design and simulation are covered in the second section, and the third section introduces the manta ray foraging optimization algorithm and Lévy-flight mechanism. The neural model structure and the training process are included in the fourth section, while the numerical results and comparisons are given in the fifth section. In the last section, discussions about the study are mentioned.

2. Design and Simulation of the C-shaped Compact Microstrip Antennas

A C-shaped compact microstrip antenna (CCMA) consists of a dielectric substrate and two thin copper planes placed on either side of this substrate, as shown in Figure 1. The C-shape was formed by removing a portion having the dimensions of *l* x *w* from the right side of the radiating copper to make the shape of the plane similar to the letter-*C*. While the radiating C-shaped copper plane was placed on top of the substrate having a dielectric constant ε_r and thickness *h*, the ground plane covered the bottom. Coaxial feeding was chosen as the feeding technique for the convenience of performing parametric simulation studies. A 50-Ω probe was used to feed the CCMA at the point of $(0.1L, 0.1W)$ for all CCMA designs.

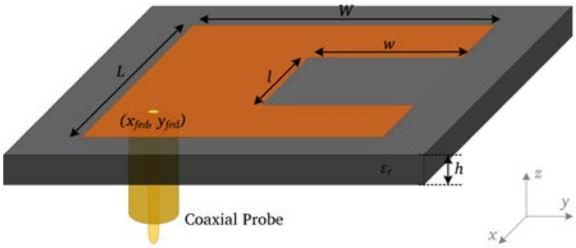


Figure 1 Three-dimensional representation of the coaxial-fed CCMA

In MA designs, the *h* parameter is generally chosen to be much smaller than the free-space wavelength (λ_0), as shown in Figure 1. In this case, the fringing field along the patch edges is minimal, while the electric field is almost normal along the patch surface [2]. Thus, as long as $h << \lambda_0$, transverse magnetic (TM) field configurations are considered for MAs, whereas with increasing operating frequency, different field configurations may arise [2]. For TM_{mnp} mode, the resonant frequency of a conventional rectangular MA (RMA) is calculated by Equation 1 [2].

$$(f_r)_{mnp} = \frac{v_0}{2\pi\sqrt{\mu_r\varepsilon_r}}\sqrt{\left(\frac{m\pi}{h}\right)^2 + \left(\frac{n\pi}{L}\right)^2 + \left(\frac{p\pi}{W}\right)^2} \tag{1}$$

In Equation 1, the indices of the radiation mode, dielectric constant, magnetic permeability, and the velocity of the electromagnetic wave in free space are represented by (m, n, p), ε_r, μ_r, and v_0, respectively. The μ_r variable defines the ferromagnetism level and can be taken as one for substrates used for the production of the MAs. Thus, in determining the resonant frequency of RMAs, the parameters *L*, *W*, *h*, and ε_r play essential roles. Since the mode of the lowest order resonant frequency is defined as the dominant mode, the mode orders are determined by placing the resonant frequencies in ascending order. When the condition $L>W>h$ is satisfied, the lowest order resonance frequency for dominant mode TM_{010} is obtained as given in Equation 2 [2].

$$(f_r)_{010} = \frac{v_0}{2L\sqrt{\varepsilon_r}} \tag{2}$$

Considering Equation 2, the boundaries of the physical parameters of CCMAs were chosen so that the resonance frequency remained within the simulated frequency band range in the dominant mode. For CCMAs, *l* and *w* parameters were critical as they had caused electrical extension along the *L*- and *W*-

axes. The outer dimensions of *L* x *W*, the slot dimensions of *l* x *w*, the substrate thickness of *h*, and the dielectric constant of the substrate ε_r were considered for CCMAs. The *L* and *l* represented the lengths, while the widths were represented by *W* and *w* parameters. For the sake of simplicity, the impact of the feeding point on the S_{11} and gain parameters was neglected. The physical and electrical parameters of the CCMA, S_{11}, and gain values for a specific frequency range were compulsory for training and testing the neural model. It was also important to choose a sufficient solution space for the proposed model to be trained. In order to achieve better results, the values should be chosen from a wide range of solutions, but simulating the entire solution area may take a long time. In this study, the values were chosen in a sufficiently wide range to ensure simplicity in simulation and training time. In order to sufficiently represent the solution space for the proposed model, the values of the thickness (*h*) and dielectric constant (ε_r) of the substrate, and the outer and inner dimensions of the radiating plane were chosen as shown in Table 1.

Table 1 Physical and electrical parameters and values of designed CCMAs

L	W	l	w	h	ε_r	Number of Simulations
30	20	3, 11, 20	3, 9, 15	1.6, 3, 4.5	2.33, 4.28, 6.15	81
45	30	4, 17, 30	4, 13, 22			81
60	40	5, 22, 40	5, 17, 30			81
75	50	6, 28, 50	6, 21, 37			81
					Total Number of Simulations	324

L, *W*, *l*, *w* and *h* parameters are in mm.

The values of *l*, *w*, *h*, and ε_r parameters shown in Table 1 were chosen to have 81 CCMA designs for each *L* x *W* pair. The height (*h*) and dielectric constant (ε_r) values were selected from the standard values provided by the substrate manufacturers. Depending on the technique-specific parameters of the FIT-based solver, each simulation session took approximately 5.1 minutes on a computer having an Intel Core i7-9750H processor and 32 GB RAM. After the simulation process, the gain and S_{11} values were extracted from the results to generate a data set. Afterward, a seven-layered DNN-based neural model was modeled by seven inputs and two outputs. Statistical calculations, including as coefficient of determination (R^2) score, mean absolute error (MAE) and mean squared error (MSE), were carried out using the gain and S_{11} values to measure training and testing performances. The followed topology is shown in Figure 2.

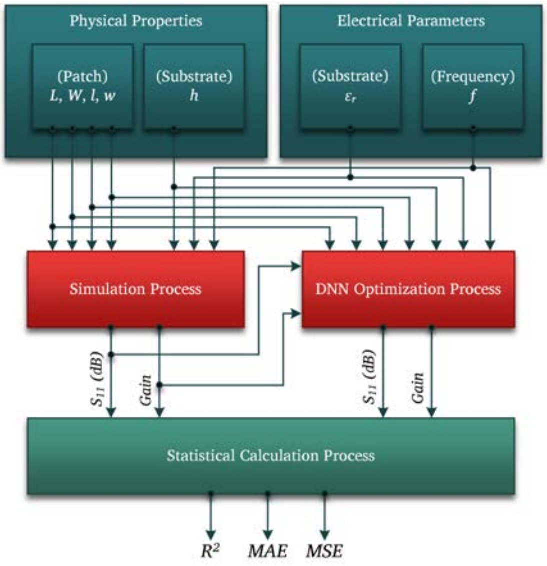


Figure 2 The topology of the processes used in the study

3. Overview of the Manta Ray Foraging Optimization (MRFO) Algorithm

The intelligent behavior of living beings has become an exciting research subject for researchers in recent years [35–37, 39, 40, 42]. One of the most recently introduced bio-inspired metaheuristic optimization techniques is the Manta Ray Foraging Optimization (MRFO) algorithm [40], which searches near-optimum solutions to complex engineering problems by imitating the intelligent foraging behaviors of manta rays. Manta rays are creatures that feed on plankton in the ocean and are highly skilled in finding abundant plankton resources. Manta rays can travel on their own as well as with groups of up to 50 members, but foraging is usually done in groups. Manta rays use three different mechanisms during the foraging phase, including chain foraging, cyclone foraging, and somersault foraging.

3.1 Chain Foraging

Chain foraging strategy is one of the foraging strategies of manta rays to explore resources that contain abundant plankton [40]. In this group based foraging strategy, manta rays are lined up head-to-tail towards the high dense plankton source. Thanks to chain sequencing, manta rays obtain the highest efficiency from the current position by allowing plankton missed by previous individuals to be caught by the next individuals in the population. The foremost manta ray travels freely to the high-density source of plankton, while other individuals follow those in front of them. The concentration level of plankton determines the position quality. In the MRFO algorithm [40], the positions of the individuals are updated by using Equation 3 according to the chain foraging strategy. Equation 3a is used to revise the position of the first individual in the population, while Equation 3b is used to update the remaining positions [40].

$$x_i^d(t+1)=\begin{cases} x_i^d(t)+r\cdot\left(x_{best}^d(t)-x_i^d(t)\right)+\alpha\cdot\left(x_{best}^d(t)-x_i^d(t)\right) & i=1 \qquad (a) \\ x_i^d(t)+r\cdot\left(x_{i-1}^d(t)-x_i^d(t)\right)+\alpha\cdot\left(x_{best}^d(t)-x_i^d(t)\right) & i=2,\dots,N \quad (b) \end{cases} \tag{3}$$

$$\alpha=2\cdot r\cdot\sqrt{|log(r)|} \tag{4}$$

In Equation 3a and 3b, the variables *x*, *r*, and α and the indices *d*, *t*, *i*, and best represent the position, a random value between (0, 1), weight coefficient, dimension, time, individual, and the highest concentrated position, respectively. The weight coefficient α used in Equation 3 is calculated using Equation 4 [40].

3.2 Cyclone Foraging

When manta ray groups discover plankton-dense positions in the deep waters, they travel to these positions with a unique strategy, called the cyclone strategy [40]. In this strategy, the manta rays are lined up as in the chain foraging strategy, and then each individual follows a spiral route while following the previous individuals. During the cyclone foraging mechanism, the positions of the individuals in the population are updated using Equation 5a for the first individual and Equation 5b for the other individuals [40].

$$x_i^d(t+1)=\begin{cases} x_{best}^d(t)+r\cdot\left(x_{best}^d(t)-x_i^d(t)\right)+\beta\cdot\left(x_{best}^d(t)-x_i^d(t)\right) & i=1 \qquad (a) \\ x_{best}^d(t)+r\cdot\left(x_{i-1}^d(t)-x_i^d(t)\right)+\beta\cdot\left(x_{best}^d(t)-x_i^d(t)\right) & i=2,\dots,N \quad (b) \end{cases} \tag{5}$$

$$\beta=2\cdot e^{r_1\frac{T-t+1}{T}}\cdot sin(2\pi r_1) \tag{6}$$

The variable β in Equation 5, and the variables *T* and r_1 in Equation 6 represent the weight coefficient, the maximum number of iterations, and a random number between 0 and 1, respectively. The exponential and sinusoidal terms in Equation 6 form the spiral route that is the basis of the cyclone strategy. By using Equation 7 and Equation 8, this foraging mechanism can be extended to allow each individual to randomly search for a better location by reference to their own best position to improve the global search performance of the MRFO algorithm [40].

$$x^d_{rand} = Lb^d + r \cdot (Ub^d - Lb^d) \tag{7}$$

$$x^d_i(t+1) = \begin{cases} x^d_{rand} + r \cdot \left(x^d_{rand} - x^d_i(t)\right) + \beta \cdot \left(x^d_{rand} - x^d_i(t)\right) & i = 1 \quad (a) \\ x^d_{rand} + r \cdot \left(x^d_{i-1}(t) - x^d_i(t)\right) + \beta \cdot \left(x^d_{rand} - x^d_i(t)\right) & i = 2, \dots, N \quad (b) \end{cases} \tag{8}$$

The lower and upper limits for the solution space of the problem are defined by the variables *L_b* and *U_b* in Equation 7. The rand in Equation 7 and Equation 8 represents the randomness. In this mechanism, the position of the first individual is updated with Equation 8a, while the positions of the other individuals are updated using Equation 8b [40].

3.3 Somersault Foraging

Another foraging strategy used by manta rays during foraging is the somersault strategy [40]. In this strategy, manta ray individuals try to pull the plankton towards the manta rays by making a series of somersaults as they circle around the position of the high-dense planktons. This mechanism is mathematically implemented as stated in Equation 9 for all individuals of the population [40].

$$x^d_i(t+1) = x^d_i(t) + S \cdot \left(r_2 \cdot x^d_{best} - r_3 \cdot x^d_i(t)\right), i = 1, \dots, N \tag{9}$$

The variables S, r_2, and r_3 in Equation 9 represent the somersault coefficient and two randomly chosen numbers between 0 and 1, respectively. Somersault coefficient S is a coefficient that defines the somersault range of manta rays and can be selected as 2 [40].

3.4 Application of the Lévy-Flight Technique to MRFO

The Lévy-flights method, proposed by the French mathematician Paul Lévy, is inspired by the natural ability that creatures in nature follow during hunting and foraging [36, 41]. When the solution quality in the current position cannot be increased during foraging, the new position is searched by moving quickly to a distant location, and this process is repeated until the best position is found [36, 41]. The Lévy-flights mechanism is used during the somersault foraging process as it offers easy applicability to the entire population in improving the best position of manta rays in the MRFO algorithm [40]. In this study, a randomly selected number was used as the decision value in implementing the Lévy-flights mechanism. The classical somersault strategy was used if the decision value had been lower than 0.5; otherwise, the Lévy-flight mechanism was used. By applying the Lévy-flight mechanism, the new positions of the individuals were calculated using Equation 10.

$$x^d_i(t+1) = Lévy^d \cdot \left(x^d_i(t) - x^d_{best}(t)\right), i = 1, \dots, N \tag{10}$$

The *$Lévy^d$* parameter given in Equation 10 represents the step size of the Lévy-flight search mechanism and is formulated in Equation 11 [36, 41].

$$Lévy^d = c_{lévy} \frac{r_4 r_5}{|r_6|^{\frac{1}{\lambda}}} \tag{11}$$

$c_{lévy}$ and λ parameters given in Equation 11 are constants, and the values of the constants were chosen as 0.001 and 1, respectively. In Equation 11, r_4, r_5, and r_6 are parameters that provide randomness. The r_4 parameter is a random number in the range of [L_b, U_b], the r_5 parameter is a random number chosen according to the Gaussian distribution in the range of [0, σ^2], and the r_6 parameter is a random number chosen according to the Gaussian distribution in the range of [0, 1] [36, 41]. As the Gamma function is a function defined as $\Gamma(x) = (x-1)!$, variance (σ^2) was calculated by the expression given in Equation 12 [36, 41].

$$\sigma^2 = \left(\frac{\Gamma(1+\lambda) \cdot sin\left(\frac{\pi\lambda}{2}\right)}{\Gamma\left(\frac{1+\lambda}{2}\right) \cdot \lambda \cdot 2^{\frac{\lambda-1}{2}}} \right)^{\frac{1}{\lambda}} \tag{12}$$

Equation 12 is based only on the constant λ, and the value of λ was chosen as 1. The pseudocode for the MRFO algorithm [40, 42], including the Lévy-flight mechanism, is shown in Algorithm 1.

Algorithm 1 Pseudocode of Manta Ray Foraging Optimization Algorithm

```
Initialize variables: population, maximum iteration number ($T_{max}$), problem dimension, lower and upper boundaries, initial positions of manta rays, and objective function
Compute objective function for initial positions and determine the best position
**REPEAT**
  **FOR** each individual
    **IF** rand(0, 1) < 0.5
      **IF** $t/T_{max}$ <rand(0, 1)
        Apply Eq. (7) and Eq. (8) for cyclone foraging
      **ELSE**
        Apply Eq. (5) and Eq. (6) for cyclone foraging
      **ENDIF**
    **ELSE**
      Apply Eq. (3) and Eq. (4) for chain foraging
    **ENDIF**
    Compute objective function and choose the best solution
    **IF** rand(0, 1) < 0.5
      Apply Eq. (9) for somersault foraging
    **ELSE**
      Apply Eq. (10) for Lévy-flight mechanism
    **ENDIF**
    Compute fitness function for each manta ray and determine the best solution
  **ENDFOR**
**UNTIL** (stop criterion is met)
```

4. Optimizing the DNN-based Model using the Improved MRFO Algorithm

The simulation study for 324 different CCMA designs with various physical and electrical values given in Table 1 was carried out utilizing a FIT-based full-wave electromagnetic solver, and the gain and S_{11} values were obtained for the desired frequency range. The S_{11} value is defined as the reflection coefficient at a specific frequency, while the gain is an important parameter that includes all the losses of the antenna. In order to obtain the gain and S_{11} values, a certain number of frequencies (n_f) must be determined between the lower frequency (f_l) and the upper frequency (f_u) in a particular frequency range. Furthermore, f_l, f_u, and n_f were chosen as 1 GHz, 5 GHz, and 70, respectively, to limit the size of the data set and obtain a sufficient solution space for the model. Studies on CCMAs generally focus on the frequency band between 1 GHz and 10 GHz in the literature [7, 29, 33, 38], which has been influential in choosing the frequency range. A seven-layered DNN-based neural model was subsequently constructed, consisting of seven inputs and two outputs, as shown in Figure 3.

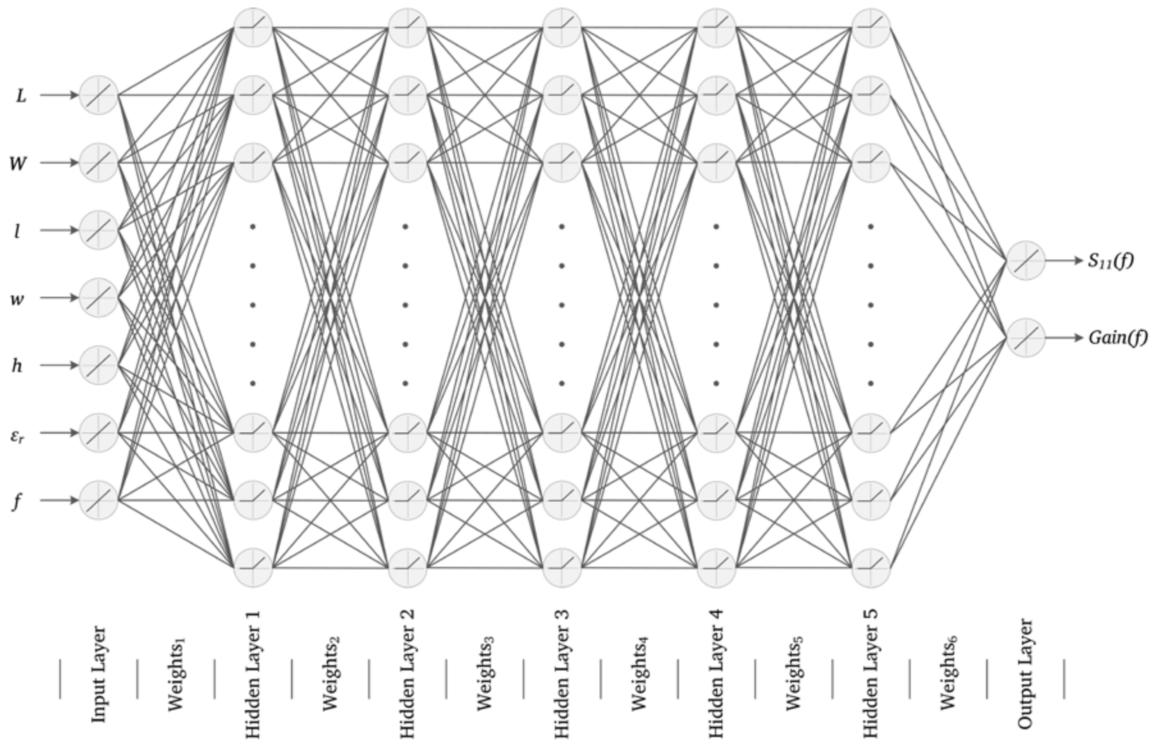


Figure 1 The proposed DNN-based model to predict the gain and S_{11} for a CCMA.

As seen in Figure 3, the proposed model consists of an input layer with seven neurons, five hidden layers with 36 neurons for each one, and an output layer with two neurons to estimate the gain and S_{11}. For each neuron in the hidden layers, the rectified linear unit activation function was used. In the training phase, data from 259 of the designed CCMAs was used, while the remaining data was used to test the model. Thus, 18144 pieces of S_{11} data were used for training the proposed model, while 4536 pieces of data were used to test the model. The training of the model can be defined as the determination of the weights and biases between layers of the DNN model that can estimate output values with the least error by using the input data set. In this study, the training of the proposed neural model was carried out using the improved MRFO algorithm [40] with the Lévy-flight mechanism [36, 41]. The weight matrices and bias vectors of the proposed neural structure constitute the problem variables of the optimization algorithm. In order to be used in the optimization process, the weight matrices and bias vectors were first transformed into a one-dimensional vector, and a single problem vector was created by combining these vectors. While the length of the problem vector determines the problem dimension, each vector element represents the problem variable to be optimized. In each iteration, the best solution computed by the algorithm was replaced by the weight matrices and bias vectors in the model, and predictions were made for the data set. Afterward, the MSE was calculated between the output values and the estimated values. The optimization algorithm continued the iterative process by searching for better solutions for the problem variables based on the mean squared error. In this study, the weight and bias matrices of the proposed model were optimized on an Intel I7-9750H CPU with 32 GB RAM. The flowchart of the optimization process is shown in Figure 4.

5. Numerical Results and Discussion

In this study, a seven-layered neural model with 5690 trainable parameters was proposed, and the model was trained during 5000 epochs with 18144 data. The training and test performances of the model were measured with the R^2 score, MAE, and MSE given in Equation 13, Equation 14, and Equation 15, respectively.

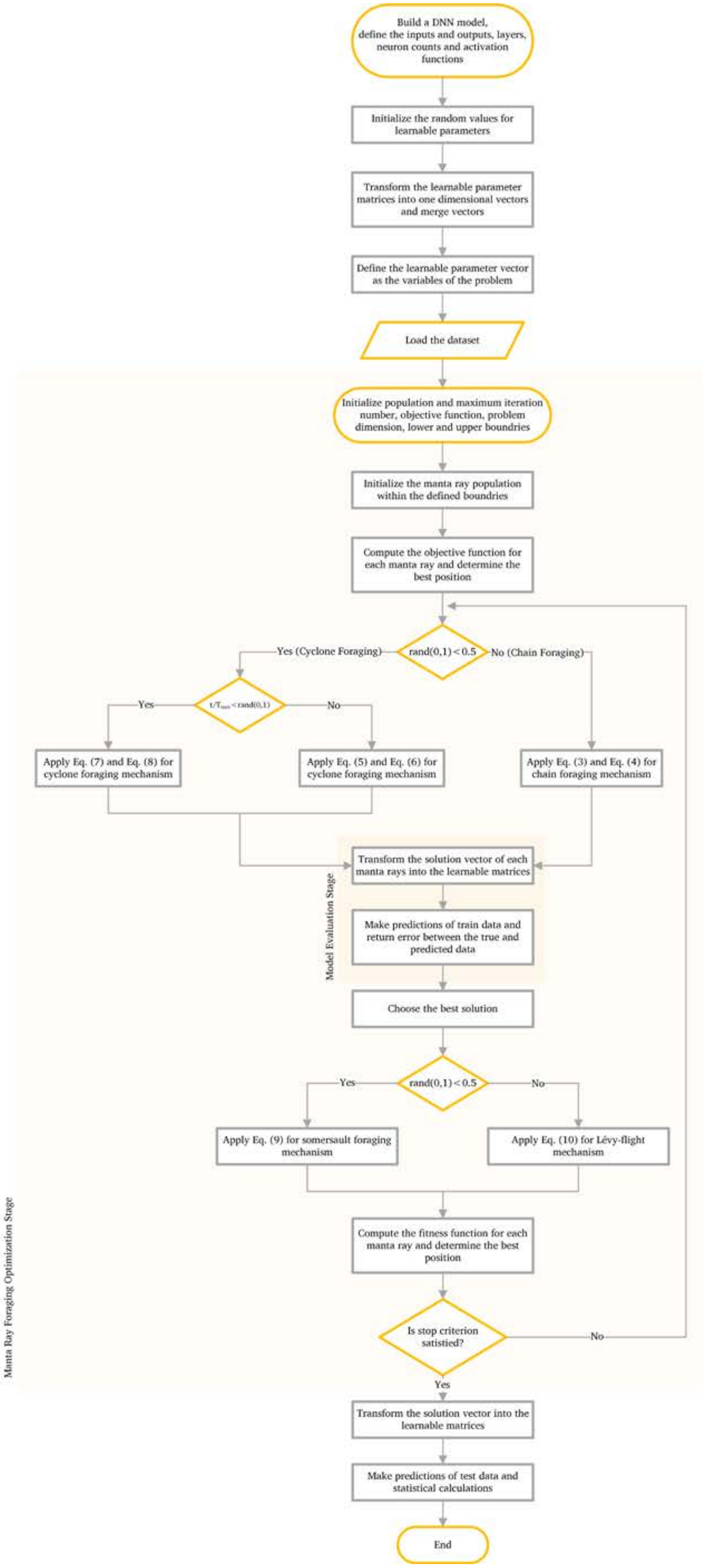

Build a DNN model, define the inputs and outputs, layers, neuron counts and activation functions
Initialize the random values for learnable parameters
Transform the learnable parameter matrices into one dimensional vectors and merge vectors
Define the learnable parameter vector as the variables of the problem
Load the dataset
Initialize population and maximum iteration number, objective function, problem dimension, lower and upper boundries
Initialize the manta ray population within the defined boundries
Compute the objective function for each manta ray and determine the best position
rand(0,1) < 0.5
Yes (Cyclone Foraging)
No (Chain Foraging)
$t/T_{max} < rand(0,1)$
Yes
No
Apply Eq. (7) and Eq. (8) for cyclone foraging mechanism
Apply Eq. (5) and Eq. (6) for cyclone foraging mechanism
Apply Eq. (3) and Eq. (4) for chain foraging mechanism
Model Evaluation Stage
Transform the solution vector of each manta rays into the learnable matrices
Make predictions of train data and return error between the true and predicted data
Choose the best solution
rand(0,1) < 0.5
Yes
No
Apply Eq. (9) for somersault foraging mechanism
Apply Eq. (10) for Lévy-flight mechanism
Compute the fitness function for each manta ray and determine the best position
Is stop criterion satistied?
No
Yes
Manta Ray Foraging Optimization Stage
Transform the solution vector into the learnable matrices
Make predictions of test data and statistical calculations
End

Figure 4 The flowchart of the MRFO-based weight and bias optimization

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_t - y_p\right)^2}{\sum_{i=1}^{n}\left(y_t - y_{t,mean}\right)^2} \tag{13}$$

$$MAE = \frac{\sum_{i=1}^{n}\left|y_t - y_p\right|}{n} \tag{14}$$

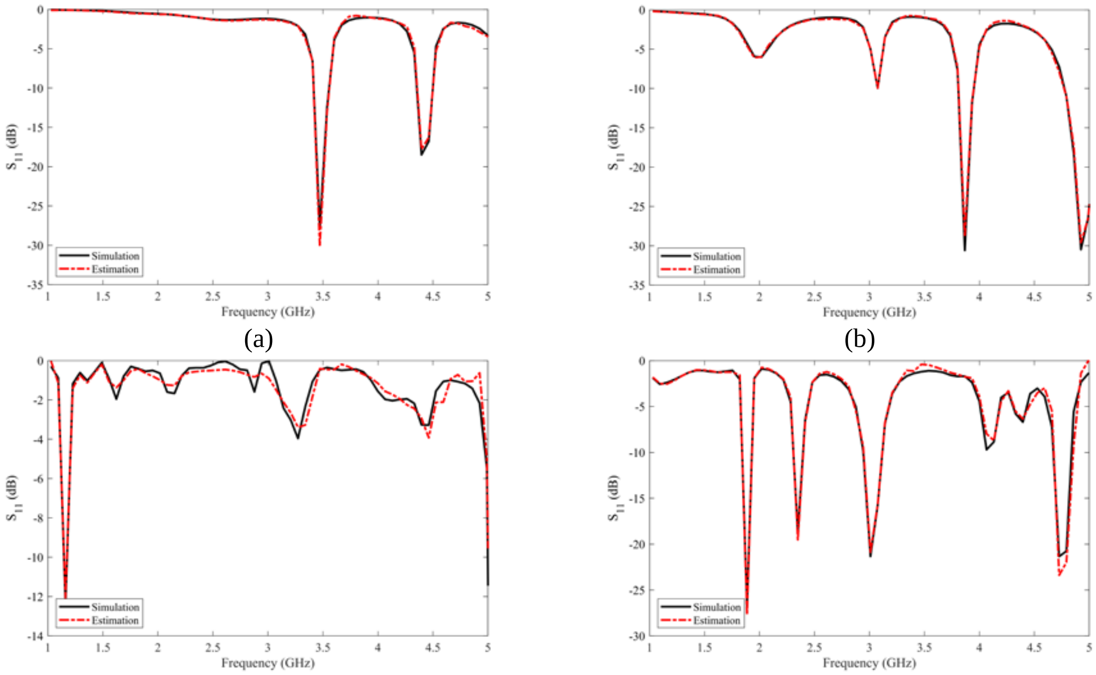
$$MSE = \frac{\sum_{i=1}^{n}\left(y_t - y_p\right)^2}{n} \tag{15}$$

In Equation 13 through 15, the terms n, y_t, y_p, and $y_{t,mean}$ represent the number of data, the target value, the predicted value, and the mean of the predicted value, respectively. The best performance metrics for training and test processes were achieved when the seed value of the random number generator was chosen as 8025443. Table 2 presents the performance metrics of the proposed model.

Table 1 Performance metrics of the proposed model for training, test, and the entire data

Metrics	Train	Test	Entire
Length of Data Set	18144	4536	22680
R^2 score	0.925	0.922	0.950
MAE	0.620	0.581	0.612
MSE	3.862	2.446	3.579

The performance metrics given in Table 2 were calculated for 70 frequency points using the physical and electrical properties of each antenna. From the table, it can be seen that the performance metrics of the proposed model for the test data are slightly better than the training data. For the current data set, the R^2 score was found to be quite close to 1, while MAE and MSE metrics were quite acceptable. According to the results, it is noted that the proposed model is not over-fitted. Therefore it can be expected to predict near-accurate results for data not included in the solution space. The comparison of the desired and predicted S_{11} curves of randomly chosen four different CCMAs operating between 1 GHz and 5 GHz is illustrated in Figure 5.



(a)

(b)

(c)

(d)

Figure 5 The simulated and estimated S_{11} curves for the randomly chosen four CCMAs, Antenna (a) #1, (b) #2, (c) #3, (d) #4 (detailed in Table 3)

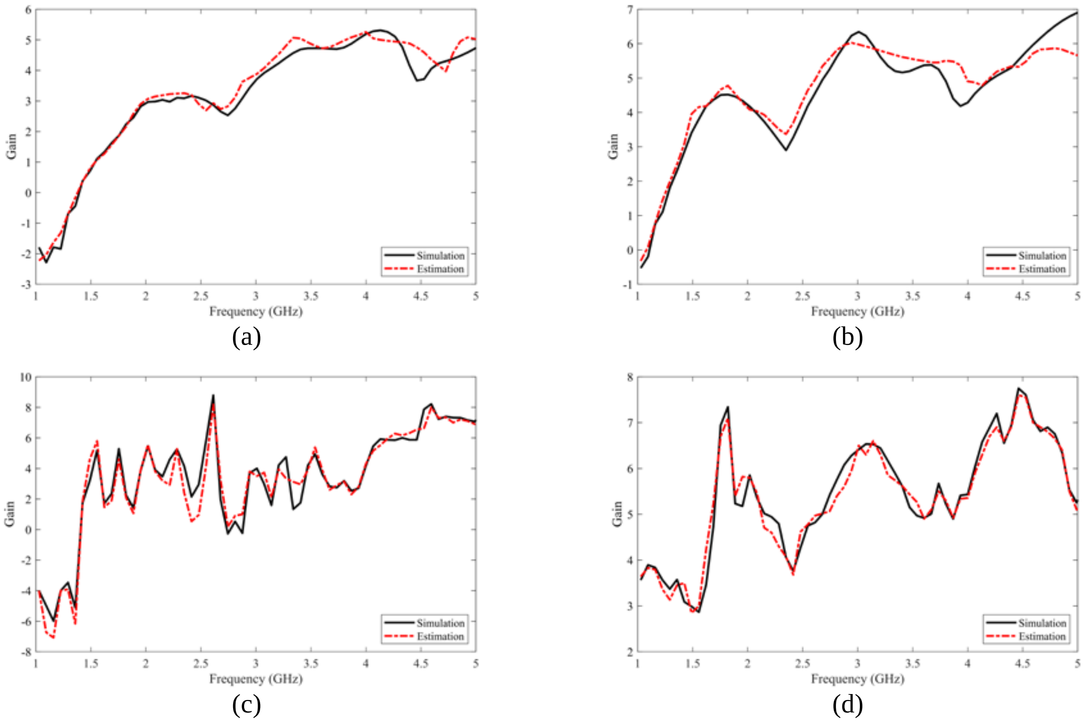
Table 2 Values of the physical and electrical parameters for the randomly chosen antennas for S_{11} curves

	Values for antennas			
Properties	**#1**	**#2**	**#3**	**#4**
L (mm)	30	45	60	75
W (mm)	20	30	40	50
l (mm)	3	17	40	28
w (mm)	9	13	30	21
h (mm)	3	4.5	1.6	4.5
ε_r	4.28	2.33	6.15	2.33
R^2	0.993	0.997	0.954	0.982

For randomly chosen four CCMAs having different outer dimensions *L* x *W*, the predicted S_{11} values between 1 GHz and 5 GHz frequencies were obtained using the model, and the target and estimated S_{11} values were compared in Figure 5 (a) – (d). The S_{11} curves illustrated in Figure 5 (a) and (b) demonstrate great harmony between the estimated S_{11} values and the target ones. As can be seen from Figure 5 (b) and (d), the increase in the values of the physical parameters causes acceptable discrepancies between the estimated and simulated values. In Figure 5 (c), the estimated values are consistent with the simulated values, but it is seen that the errors have increased slightly. The comparative curves for simulated and estimated gain values for four randomly chosen CCMAs are given in Figure 6.

Table 3 Values of the physical and electrical parameters for the randomly chosen antennas for gain curves

	Values for antennas			
Properties	**#1**	**#2**	**#3**	**#4**
L (mm)	30	45	60	75
W (mm)	20	30	40	50
l (mm)	3	17	40	28
w (mm)	9	13	30	21
h (mm)	3	4.5	1.6	4.5
ε_r	4.28	2.33	6.15	2.33
R^2	0.982	0.968	0.995	0.996



(a)

(b)

(c)

(d)

Figure 6 The simulated and estimated gain curves for the randomly chosen four CCMAs, Antenna (a) #1, (b) #2, (c) #3, (d) #4 (detailed in Table 4)

As shown in Figure 6, the model made relatively more errors for the estimation of the gain values than the S_{11} values. Due to the use of a single model with two outputs and the limited number of data, acceptable mismatches occurred between the estimated data and the simulation data. As seen in Figure 5 and Figure 6, while the harmony is high at low frequencies, the discrepancy between the estimated data and the simulation data increases as the frequency increases. As mentioned in the literature review, most of the studies related to CCMAs have focused on optimizing the antenna designs for specific applications and obtaining mathematical models that usually provide the resonant frequency [3, 5, 6, 17–34]. Although this study is not directly related to the estimation of resonant frequencies of CCMAs, the resonant frequencies in dominant mode (TM_{010}) were indirectly estimated by the interpretation of the S_{11} curves, and the comparison between the results obtained in this study and the results in the literature [7, 23] is shown in Table 5. Comparisons of resonant frequencies are made based on the simulation results presented by [5].

Table 5 Comparative results and percentage errors on resonant frequencies

#	Slot dimensions (mm)		Resonant frequencies (GHz)							Percentage errors (%)					
			Simulated	Estimated	Calculated										
	l	w	[5]	This Study	[22]	[6]	[5]			This Study	[22]	[6]	[5]		
							#1	#2	#3				#1	#2	#3
1	5	5	1.562	1.559	1.562	1.657	1.502	1.630	–	0.192	0.000	6.082	3.841	4.353	–
2	10	10	1.445	1.452	1.426	1.497	1.398	1.408	–	0.484	1.315	3.599	3.253	2.561	–
3	15	15	1.286	1.271	1.280	1.334	1.309	1.241	–	1.166	0.467	3.732	1.788	3.499	–
4	20	20	1.130	1.102	1.136	1.178	1.231	1.111	1.002	2.478	0.531	4.248	8.938	1.681	11.327
5	25	25	0.991	0.998	1.000	1.035	1.164	1.008	0.928	0.706	0.908	4.440	17.457	1.715	6.357
6	40	30	0.899	0.892	0.890	0.924	–	0.893	0.856	0.779	1.001	2.781	–	0.667	4.783
7	5	30	0.929	0.916	0.931	0.963	–	1.029	0.904	1.399	0.215	3.660	–	10.764	2.691
8	10	30	0.887	0.881	0.911	0.938	–	–	0.896	0.676	2.706	5.750	–	–	1.015
9	2	30	0.964	0.959	0.946	0.982	–	–	0.910	0.519	1.867	1.867	–	–	5.602
Average percentage errors (APE)										0.933	1.001	4.018	7.055	3.606	5.296

L=60 mm, W = 40 mm, h = 1.59 mm, ε_r = 2.33
– Not available.

The physical dimensions of antennas between #1 – #8 in Table 5 are within the range of dimensions, while #9 is outside the range. The dominant resonant frequencies estimated using the proposed model, however, were entirely in line with the literature, and the percentage error (PE) values were calculated to be considerably small. For literature data, the average PE (APE) value was calculated as 0.933% for the estimated resonant frequencies. Afterward, a CCMA with outer dimensions 30 x 20 mm^2 and slot dimensions 3 x 7 mm^2 was fabricated using Rogers RT/duroid 5870 substrate with a thickness of 1.57 mm [23]. The antenna was fed from the bottom-left concerning the *L*-axis using a 90-degree SMA connector. The S_{11} of the CCMA was measured by using Agilent E5071B ENA RF Series Network Analyzer. The estimated S_{11} curve using the proposed model was compared with the simulated and measured values of the fabricated antenna, and the comparative curves are shown in Figure 7.

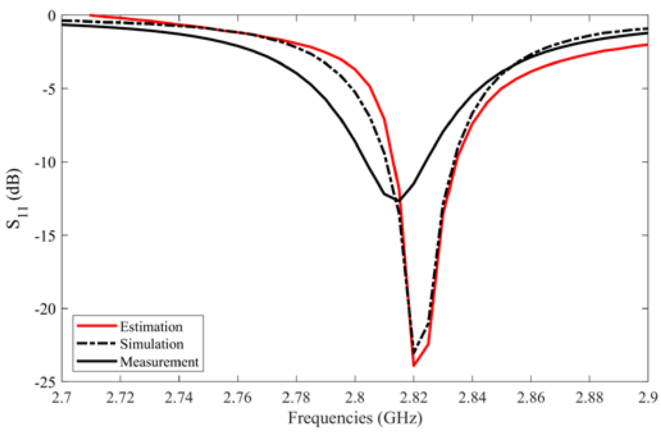


Figure 7 Comparison of the S_{11} curves for the CCMA (ε_r = 2.33)

While the S_{11} curve obtained in the simulation and the estimated S_{11} curve obtained by the model are in harmony, the measured S_{11} curve of the fabricated CCMA is similar in shape but shifts slightly to the left of the frequency axis, as shown in Figure 7. Since the model is trained using the data set obtained from the simulation, it yielded a similar result to the data obtained by the simulation, though with slight inaccuracy. The training data set needs to be expanded, and the number of layers in the model needs to be increased in order to achieve better results. However, since the study aimed to ensure that the training data set is precisely learned by the model, the result shown in Figure 7 is given only for the purpose of benchmarking.

6. Conclusion

Recent studies in the literature have focused on estimating the characteristics of microstrip antennas, such as resonant frequencies and gains, using machine learning methods. This study was focused on the estimation of S_{11} and gain values in a certain frequency range instead of estimating the resonant frequencies of microstrip antennas. For this purpose, a DNN-based neural model was proposed and optimized using the MRFO algorithm. The gain and S_{11} values of CCMAs, operating within the frequency range from 1 GHz to 5 GHz, were obtained for specific physical and electrical properties to build the data set. The weight matrices and bias vectors of the proposed model were optimized using this data set. R^2 score that shows the success of the model for training and test data was obtained as 0.925 and 0.922, respectively. Besides, the average percentage error (APE) for the prediction of resonant frequency between the obtained data with the literature data was calculated as 0.933%. It is seen that extending the data set by making more antenna designs having various physical and electrical characteristics and by making more simulations for smaller frequency steps in the wider frequency band will increase the performance of the model. The results show that DNN-based models can be utilized successfully in estimating the gain and S_{11} values of CCMAs. The proposed model gives fast and practical results, and it can be used as a computer-aided design (CAD) tool by antenna designers and researchers.

Authors' Contributions

All authors contributed equally to the study.

Statement of Conflicts of Interest

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The author declares that this study complies with Research and Publication Ethics.

References

[1] J. Q. Howell, "Microstrip Antennas," *IEEE Transactions on Antennas and Propagation*, vol. 23, pp. 90–93, 1975.

[2] C. A. Balanis, *Antenna Theory: Analysis and Design*. Wiley & Sons, Inc., 2016.

[3] R. N. Tiwari, P. Singh, and B. K. Kanaujia, "A modified microstrip line fed compact UWB antenna for WiMAX/ISM/WLAN and wireless communications," *AEU - International Journal of Electronics and Communications*, vol. 104, pp. 58–65, 2019.

[4] G. Kumar, and K. P. Ray, *Broadband Microstrip Antennas*. Artech House, 2003.

[5] A. A. Deshmukh, and G. Kumar, "Formulation of resonant frequency for compact rectangular microstrip antennas," *Microwave and Optical Technology Letters*, vol. 49, no. 2, pp. 498–501, 2007.

[6] A. H. Yuzer, and C. Seker, "4G C-shaped compact microstrip antenna design and production," *Pamukkale University Journal of Engineering Sciences*, vol. 23, no. 5, pp. 532-535, 2017.

[7] A. Akdagli, M. B. Bicer, and S. Ermis, “A novel expression for resonant length obtained by using artificial bee colony algorithm in calculating resonant frequency of C-shaped compact microstrip antennas,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 19, no. 4, pp. 597–606, 2011.

[8] A. K. Bhattacharyya, and R. Garg, “Generalized transmission line model for microstrip patches,” *Microwaves Optics and Antennas*, vol. 132, no. 2, pp. 93–98, 1985.

[9] W. Richards, Y. Lo, and D. Harrison, “An improved theory for microstrip antennas and applications,” *IEEE Transactions on Antennas and Propagation*, vol. 29, no. 1, pp. 38–46, 1981.

[10] A. Taflove, and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Artech House, 2005.

[11] S. C. Gao, L. W. Li, M. S. Leong, T. S. Yeo, “Analysis of an H-shaped patch antenna by using the FDTD method,” *Progress In Electromagnetics Research*, vol. 34, pp. 165–187, 2001.

[12] J. L. Volakis, A. Chatterjee, and L. C. Kempel, *Finite Element Method Electromagnetics: Antennas, Microwave Circuits, and Scattering Applications*. Wiley, 1998.

[13] T. Weiland, and M. Clemens, “Discrete electromagnetism with the finite integration technique,” *Progress In Electromagnetics Research*, vol. 32, pp. 65–87, 2001.

[14] T. Weiland, “A discretization method for the solution of Maxwell’s equations for six-component fields,” *AEU - International Journal of Electronics and Communications*, vol. 31, no. 3, pp. 116–120, 1977.

[15] R. F. Harrington. *Field Computation by Moment Methods*. Wiley, 1993.

[16] E. H. Newman, and P. Tulyathan, “Analysis of microstrip antennas using moment methods,” *IEEE Transactions on Antennas and Propagation*, vol. 29, no. 1, pp. 47 – 53, 1981.

[17] X. Cao, Y. Xia, L. Wu, and L. Lang, “A compact quad-band ring string-shaped antenna,” *AEU - International Journal of Electronics and Communications*, vol. 111, pp. 152910, 2019.

[18] A. T. Mobashsher, K. S. Bialkowski, and A. M. Abbosh, “Design of compact cross-fed three-dimensional slot-loaded antenna and its application in wideband head imaging system,” *IEEE Antennas and Wireless Propagation Letters*, vol. 15, pp. 1856 – 1860, 2016.

[19] D. Ustun, and A. Akdagli, “Design of a dual-wideband monopole antenna by artificial bee colony algorithm for UMTS, WLAN, and WiMAX applications,” *International Journal of Microwave and Wireless Technologies*, vol. 9, no. 5, pp. 1197–1208, 2017.

[20] J. M. Felicio, J. M. Bioucas-Dias, J. R. Costa, and C. A. Fernandes, “Antenna design and near-field characterization for medical microwave imaging applications,” *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 7, pp. 4811 – 4824, 2019.

[21] A. Akdagli, C. Ozdemir, S. Yamacli, and C. C. Arcasoy. “Improved formulas for the resonant frequencies of dual frequency arrow shaped compact microstrip antenna,” *Microwave and Optical Technology Letters*, vol. 50, no. 1, pp. 62–65, 2008.

[22] V. S. Chintakindi, S. S. Pattnaik, O. P. Bajpai, S. Devi, P. K. Patra, and K. M. Bakwad, “Resonant frequency of equilateral triangular microstrip patch antenna using particle swarm optimization technique,” *2008 International Conference on Recent Advances in Microwave Theory and Applications*, 2008.

[23] A. Toktas, M. B. Bicer, A. Akdagli, and A. Kayabasi, “Simple formulas for calculating resonant frequencies of C and H shaped compact microstrip antennas obtained by using artificial bee colony algorithm,” *Journal of Electromagnetic Waves and Applications*, vol. 25, no. 11-12, pp. 1718–1729, 2011.

[24] M. Biswas, and M. Dam, “Closed-form model to determine the co-axial probe reactance of an equilateral triangular patch antenna,” *International Journal of Microwave and Wireless Technologies*, vol. 10, no. 7, pp. 801 – 813, 2018.

[25] B. Sami, B. Siham, and F. Tarek, “Analysis of a circular microstrip antenna on isotropic or uniaxially anisotropic substrate using neurospectral approach,” *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 33, no. 1/2, pp. 567–580, 2014.

[26] L. Barkat, S. Bedra, T. Fortaki, and R. Bedra, “Neurospectral computation for the resonant characteristics of microstrip patch antenna printed on uniaxially anisotropic substrates,” *International Journal of Microwave and Wireless Technologies*, vol. 9, no. 3, pp. 613-620, 2017.

[27] A. Kayabasi, A. Toktas, A. Akdagli, M. B. Bicer, and D. Ustun, "Applications of ANN and ANFIS to predict the resonant frequency of L-shaped compact microstrip antennas," *Applied Computational Electromagnetics Society Journal*, vol. 29, no. 6, 2014.

[28] A. Kayabasi, M. B. Bicer, A. Akdagli, and A. Toktas, "Computing resonant frequency of H-shaped compact microstrip antennas operating at UHF band by using artificial neural networks," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 26, no. 4, 2011.

[29] D. Ustun, A. Toktas, and A. Akdagli, "Deep neural network–based soft computing the resonant frequency of E–shaped patch antennas," *AEU - International Journal of Electronics and Communications*, vol. 102, pp. s54-61, 2019.

[30] M. Kaur, and J. S. Sivia, "Giuseppe Peano and Cantor set fractals based miniaturized hybrid fractal antenna for biomedical applications using artificial neural network and firefly algorithm," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 30, no. 1, 2020.

[31] K. Y. Kapusuz, H. Tora, and S. Can, "Neural network based estimation of resonant frequency of an equilateral triangular microstrip patch antenna," *Tehnicki Vjesnik*, vol. 20, no. 6, pp. 955-960, 2013.

[32] S. Can, K. Y. Kapusuz, and E. Aydin, "Neural network based resonant frequency solver for rectangular-shaped shorting pin-loaded antennas," *Microwave and Optical Technology Letters,* vol. 55, no. 12, pp. 3025-3028, 2013.

[33] J. Singh, G. Singh, S. Kaur, and B. S. Sohi, "Performance analysis of different neural network models for parameters estimation of coaxial fed 2.4 GHz E-shaped microstrip patch antenna," *2015 2nd International Conference on Recent Advances in Engineering and Computational Sciences*, 2015.

[34] T. M. Neebha, and M. Nesasudha, "Artificial neural network based design of a microstrip patch antenna for RADAR applications," *Proceedings of the International Conference on Recent Advances in Aerospace Engineering*, 2017.

[35] D. Karaboga, and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[36] X. S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley, 2010.

[37] A. Kaveh, and T. Bakhshpoori, *Metaheuristics: Outlines, MATLAB Codes and Examples*. Springer, 2019.

[38] P. K. Abbassi, N. M. Badra, A. M. M. A. Allam, and A. El-Rafei, "WiFi antenna design and modeling using artificial neural networks," *Proceedings of 2019 International Conference on Innovative Trends in Computer Engineering,* 2019.

[39] X. S. Yang, "Firefly algorithms for multimodal optimization," *Lecture Notes in Computer Science*, 2009.

[40] W. Zhao, Z. Zhang, L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Engineering Applications of Artificial Intelligence,* vol. 87, pp. 103300, 2020.

[41] Y. Li, X. Li, J. Liu, and X. Ruan, "An improved bat algorithm based on lévy flights and adjustment factors," *Symmetry*, vol. 11, no. 7, pp. 925, 2019.

[42] O. E. Turgut, "A novel chaotic manta-ray foraging optimization algorithm for thermo-economic design optimization of an air-fin cooler," *SN Applied Sciences*, vol. 3, no. 1, pp. 3, 2021.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 4, NO. 2, AUGUST 2021

Research Article

Deep Gated Recurrent Unit for Smartphone-Based Image Captioning

Volkan Kılıç[1]
[1]Department of Electrical and Electronics Engineering, Izmir Katip Celebi University, Izmir, 35620, Turkey;
volkan.kilic@ikcu.edu.tr; +90 232 329 35 35/3881

Received 22 January 2021; Revised 11 May 2021; Accepted 13 May 2021; Published online 31 August 2021

Abstract

Expressing the visual content of an image in natural language form has gained relevance due to technological and algorithmic advances together with improved computational processing capacity. Many smartphone applications for image captioning have been developed recently as built-in cameras provide advantages of easy-operation and portability, resulting in capturing an image whenever or wherever needed. Here, an encoder-decoder framework based new image captioning approach with a multi-layer gated recurrent unit is proposed. The Inception-v3 convolutional neural network is employed in the encoder due to its capability of more feature extraction from small regions. The proposed recurrent neural network-based decoder utilizes these features in the multi-layer gated recurrent unit to produce a natural language expression word-by-word. Experimental evaluations on the MSCOCO dataset demonstrate that our proposed approach has the advantage over existing approaches consistently across different evaluation metrics. With the integration of the proposed approach to our custom-designed Android application, named "*VirtualEye+*", it has great potential to implement image captioning in daily routine.

Keywords: artificial intelligence, natural language processing, image captioning, Android

1. Introduction

The problem of image captioning has received much attention from the computer vision (CV) and natural language processing (NLP) communities in recent decades due to its potential applications including image indexing or retrieval, virtual assistants for visually impaired people [1, 2]. Image captioning needs a higher level of image understanding beyond object detection and classification to identify the objects and actions which plays a critical role to generate expressions for an image in the form of a natural language with proper linguistic properties. Early efforts in image captioning often use either template-based [3-7] methods or retrieval-based [8-11] methods. The template-based methods employ image information such as objects, scenes and attributes to generate a meaningful caption using the most relevant words from sentence templates. The captions have constant length and highly sensitive to the performance of the object detector which leads to simple sentences with a tendency to deviate from the ground truth captions. To address these problems, the visual information of the input image was employed in the retrieval-based methods to match ground truth captions of the most likely images from the retrieval library. The matched ground truth captions are collected to get more flexible and semantically richer captions. The main drawback in the retrieval-based methods is that the generated captions may be misleading if similar images are not contained in the retrieval library.

These drawbacks in the template-based and retrieval-based methods have been overcome with neural network based methodologies which combine convolutional neural network (CNN) and recurrent neural network (RNN) [12, 13]. The problem of image captioning is formulated as a translation problem [14] inspired by machine translation [15, 16]. The CNN and RNN methods are used under the encoder-decoder framework [17-19] which leads to employing deep networks in image captioning. The encoder consists of deep CNNs used to extract visual information from an image. Recently, advanced CNN architectures including NASNetLarge [20], Xception [21] and Inception-v3 [22] have emerged which show promising performance under various encoder designs. The RNN-based decoder converts the extracted information by CNN-based encoders into natural language captions word-by-word. Conventional RNNs, however, have vanishing and exploding gradient problems which prevents to employ of sufficiently long-term temporal dependencies [13, 23, 24]. To address these issues, long short-term memory (LSTM) [25] and gated recurrent unit (GRU) [26] networks are proposed. The

To Cite This Article: V. Kilic, "Deep Gated Recurrent Unit for Smartphone-based Image Captioning," Sakarya University Journal of Computer and Information Sciences, vol. 4, no. 2, pp. 181–191, 2021. doi: 10.35377/saucis.04.02.866409

LSTM uses a memory cell to store information for long periods of time in memory while GRU keeps the flow of information without additional memory cells. Chen et al. [27] proposed a style-factual image caption generator that uses the encoder-decoder framework with LSTM and they injected the style-factual features into the decoder. The factual representation is obtained by utilizing an adaptive learning approach. You et al. [28] presented an image caption generator where they employed the encoder-decoder framework with LSTM. Term generator and language generator were proposed in [29] where the CNN features of images are fed with terms from reference captions into the initial hidden state of the GRU. The term generator produces words which are the input of the language generator for the final caption.

RNN-based decoders generally process visual information from the encoder under two approaches [30]. The first approach includes direct feeding from the encoder into the RNN while the latter approach uses one additional layer before the RNN. These approaches can be sorted as init-inject, pre-inject, par-inject and merge architectures [30]. The init-inject receives the visual information as an image vector to feed to the initial hidden state of the RNN [31]. The representative study in this category is the scene graph [31] which detects the objects and extracts attributes to feed into the initial hidden state vector of the LSTM based RNN together with the CNN image features. In pre-inject architecture, the RNN takes the image vector as a first input [32-35] while the par-inject architecture employs the image vector and word vectors of the caption prefix in parallel as an input to the RNN [14, 36]. The merge-architecture employs the image after the RNN generates the caption prefix instead of feeding the image vector directly to the RNN [12, 37, 38]. The comparison of these architectures was reported in [30] that the init-inject shows superior performance in terms of generation and retrieval measures.

In this study, a neural encoder-decoder framework based new image captioning model is proposed which encodes the images with Inception-v3 CNN to generate captions using multi-layer GRU based RNN decoder under init-inject architecture. Among the comparison of NASNetLarge, Xception and Inception-v3 on the proposed image captioning model, Inception-v3 outperforms the others in terms of encoding visual information. In RNN based decoder, GRU is employed because of its computational efficiency and simplicity as it includes one hidden state vector while LSTM is operated with two state vectors consists of hidden and cell states [39]. Moreover, LSTM has input, forget and output gates while GRU uses only two gates: update and reset. In terms of compatibility with init-inject architecture, GRU with one hidden state vector also offers the best choice [30]. The motivation behind using multiple layers in GRU is to utilize the most relevant information in the unit which improves the ability of the decoder in utilizing visual information, so that an enhanced prediction model for caption generation is provided [40]. The proposed model is then compared with other existing approaches for caption generation, using performance metrics such as BLEUn (with n = 1, …, 4) [41], METEOR [42], CIDEr [43], and ROUGEL [44]. Typically, BLEUn is a type of n-grams precision measure as BLEU3 means 3-grams. METEOR is the harmonic mean of the precision and recall score for unigrams while CIDEr measures the similarity of n-grams using weighted cosine function, and ROUGEL is a score of longest common subsequence [45].

The rest of the paper is organized as follows: Section 2 introduces the theoretical foundations for the encoder and decoder framework. The proposed image captioning approach is given in Section 3. Experimental results performed on the MSCOCO dataset and performance comparison of the approaches are discussed in Section 4, followed by the conclusions.

2. Encoder-Decoder Framework for Image Captioning

The theoretical foundation of our proposed approach for image captioning based on the encoder-decoder framework is described in this section. The CNN architectures employed to obtain visual features and attributes of the image are introduced before the RNN-based decoders which are used to produce the image captions.

2.1 Encoder

Encoding an image means converting image data into a feature vector that contains the image information. Conventional encoder designs are based on a CNN due to its ability to deal with high dimensional data and remarkable feature extraction capability. In CNN, there are convolutional, pooling, and fully connected layers. In the convolutional layer, a filter is convolved with an image to create a feature or activation map which contains the detected features in the image. The pooling layer is a sampling layer that gradually decreases the spatial size of the feature map, resulting in a reduced number of features and computational complexity [46]. The fully connected layer produces the final decision based on all input from the previous layers.

Image captioning requires advanced computer vision techniques for image analysis and feature extraction. Deep CNN architectures best fit the requirement of convenient feature extraction that improves the quality of the captions. Therefore, pre-trained deep CNN architectures are used in this study such as NASNetLarge, Xception, and Inception-v3 in the encoder part.

Inception-v3 is a deep CNN, which is consists of 42-layers of convolutional, pooling, and fully connected layers. This architecture procured second place in ILSVRC 2015. Xception is a novel deep CNN obtained by modifying Inception-v3 with depth-wise sectional convolutions. Therefore, Xception architecture surpassed Inception-v3 on the ImageNet dataset. NASNetLarge (Neural Architecture Search Network) is a constructed CNN architecture designed using reinforcement learning on the CIFAR-10 dataset [47]. The ImageNet dataset is used in training of the architecture which leads to state-of-the-art performance.

The encoder extracts a high-level feature from the image using the convolutional and pooling layers of the CNN architectures. Then, the features are fed into the decoder for caption generation.

2.2 Decoder

A decoder produces words to describe an image with semantically meaningful sentences by using feature representation. Decoders are mostly designed based on RNN as it is capable of storing parts of the inputs and use them to generate meaningful captions.

RNN is a type of deep network that uses its internal state to process input sequences, which makes it suitable for sequential applications including speech recognition and image captioning [46, 48]. RNN calculates each output employing the same function over each instance of the sequence repeatedly. RNN consists of a hidden state and an optional output which operate on the input sequence. The current hidden state has been computed by taking the current input with the hidden state for the former time step using a nonlinear activation function which leads to an update of the output at each time step. The motivation behind using the RNN relies on the generalization of the solution with respect to time and its capability to deal with sequences which a classical deep learning architecture can not be applied directly. However, RNN suffers from the problem of vanishing and exploding gradients. Therefore, it cannot maintain long term dependencies. This problem is addressed by employing GRU which is a type of RNNs with a gating mechanism.

Conventional GRU with a hidden state, update and reset gates is depicted in Figure 1.

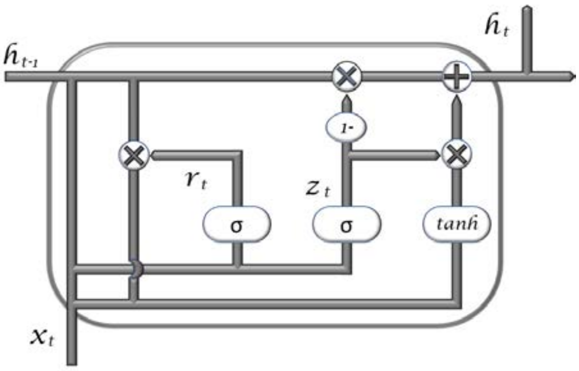


Figure 1 Gated Recurrent Unit

In GRU, the flow of information is maintained with following equations [49]:

$$r_t = \sigma(W_{xr}x_t + U_{hr}h_{t-1}) \quad (1)$$

$$z_t = \sigma(W_{xz}x_t + U_{hz}h_{t-1}) \quad (2)$$

$$u_t = tanh(W_{xu}x_t + U_{hu}(r_t \odot h_{t-1})) \quad (3)$$

$$h_t = (1 - z_t)h_{t-1} + z_t u_t \quad (4)$$

where $\mathrm{x_t}$ and $\mathrm{h_t}$ are the input and output vectors, respectively. Reset gate vector is defined with $\mathrm{r_t}$ and $\mathrm{z_t}$ denotes the update gate. The tangent hyperbolic activation function is denoted with tanh while σ is the sigmoid functions. Parameters are defined with W and U matrices, and ⊙ defines element-wise multiplication. The vanishing gradient problem is handled by a gating mechanism in the GRU while the exploding gradient is addressed with gradient clipping strategy [50]. The input is taken from the previous layer at each time step which provides to configure the GRU with multiple layers, resulting in outstanding performance compared to the conventional RNN-based architectures on many NLP tasks, including language modeling [51, 52].

3. Proposed Image Captioning Approach

This section presents a new approach to improve the image captions by introducing multi-layer GRU into an RNN-based decoder. After the proposed approach is described in the next subsection, our custom-designed Android application which runs the proposed approach under a user-friendly interface is demonstrated.

3.1 Deep Gated Recurrent Unit for Image Captioning

The encoder-decoder framework comprises CV and NLP algorithms in the encoder and decoder, respectively. CNN based encoders implement the CV tasks by extracting feature representation of an image while RNN-based decoders translate this representation into natural language captions. Image features and linguistic features can be deployed in RNN using different types of architectures, such as init-inject, pre-inject, par-inject and merge architectures [30]. The feature vector of the image has been employed as RNN initial hidden state vector in init-inject architecture. The image feature vector and the hidden state vector of the RNN should be the same size to meet the requirement of an early binding architecture which helps the RNN to change the image representation. It is noted in [30] that the init-inject outperforms the others in terms of generation and retrieval measures. In this study, a new deep GRU design is proposed in an RNN-based decoder under init-inject architecture for natural language descriptions of the image.

The queried image features are processed to generate a caption by the decoder involving an embedding layer, GRUs, and a fully connected (FC) layer. The proposed RNN-based decoder with multi-layer GRU is given in Figure 2. The GRU learns how to process image features and vectors to generate the most meaningful attributions. The embedding layer represents words as meaningful vectors. The fully connected layer predicts the most applicable word corresponding to the attributions.

As CNNs are not capable of handling word sequences, the conversion of words to vectors is needed to process in the RNN. Word embedding is the common approach to obtain vectors that contain semantics of the corresponding words. Here, the words were indexed into integer-tokens and converted into 128-sized float arrays by using an embedding layer. The embedding layer was trained along with the network to capture the more compact features of the words.

The captions were considered as time-series data of words and GRUs were utilized to learn the connection between the words in a caption. The four-layer GRU was constructed by combining the four individual RNN with their initial states. All features from the encoder were split into four equal-sized vectors, and each vector was fed to the initial state of GRU layers sequentially. Time series data from the output of the embedding layer was employed as input for the GRU layers. The correct token is predicted by a fully connected layer at the end of the decoder for semantically more meaningful captions.

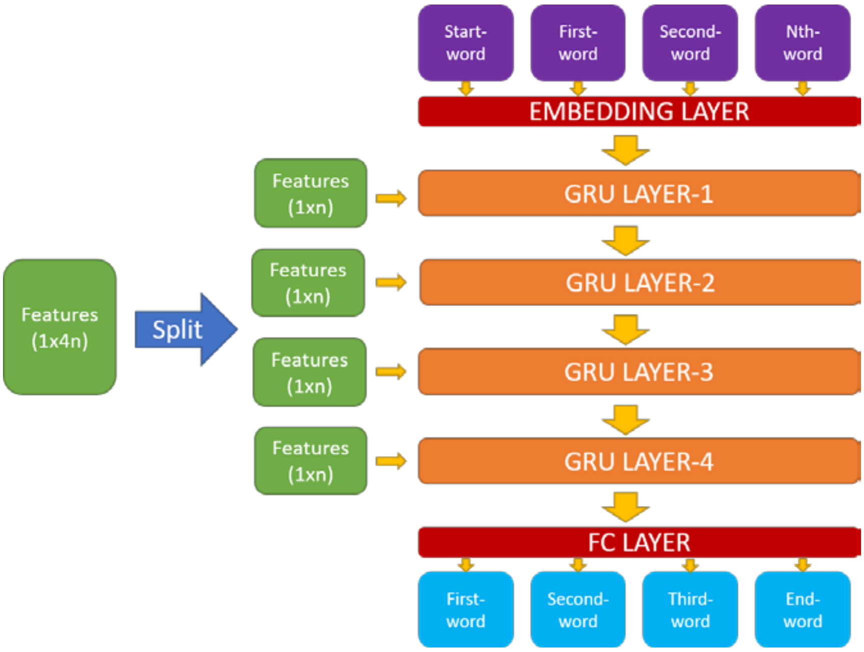


Figure 2 Multi-layer GRU based Decoder

3.2 Smartphone Application: *Virtual Eye+*

Our previous custom-design Android application *Virtual Eye* [53] was upgraded with new features and named it *Virtual Eye+* which provides more simplicity and easy operation.

The improvements on the application can be sorted into three categories; user interface, server, and cloud communication. First, the user interface was improved to increase the capability of easy-operation and portability, so that it can be benefited whenever captioning is needed. The homepage provides a manual that introduces the application for image captioning. In addition, “choose file” and “generate caption” buttons are given on the home screen. An image can be chosen from the gallery by tapping the “choose file” button. When the “generate caption” button is tapped, the application sends the selected image to the server and waits for the caption. The received caption is displayed on the homepage under the input image and it can be audible when the input image is tapped. Moreover, an in-app camera is provided where the user can access it by scrolling left from the homepage. An image can be captured by tapping anywhere on the screen and then, *Virtual Eye+* automatically sends the photo to the server and scrolls back to the homepage. The generated caption on the server is English as a default. However, it is possible to change language from the settings of the *Virtual Eye+* and so that the English caption can be translated to the smartphone display language.

Second, the server was set up on the Kivy user interface which generates a caption for the input image. If the user has requested a “non-English” caption, the caption is translated using the translate API from google before sending it to the user. Lastly, cloud communication is fulfilled under the Firebase cloud service which allows fast and robust communication between Python-based server and Java-based Android application. When an image is uploaded to the Firebase storage, its download link is activated which invokes the server to download the image via the link. The overall system is demonstrated in Figure 3.

4. Experimental Evaluations

4.1 Dataset and Performance Metrics

In order to evaluate the proposed captioning approach and compare it with existing approaches, a dataset including a large number of images with reference captions is required.

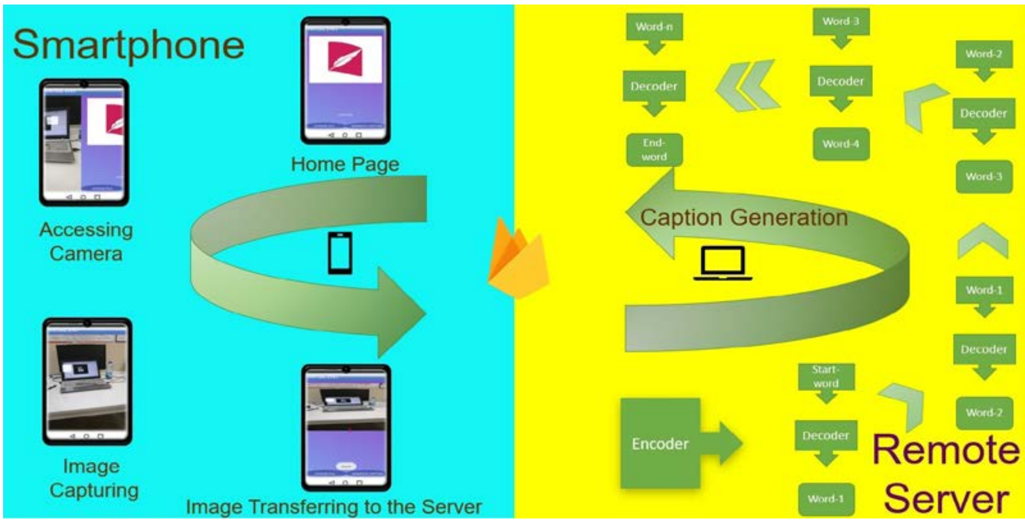


Figure 3 The Working Principle of Virtual Eye+

Apart from MSCOCO [54], the suitability of several other publicly available captioning datasets such as Flickr [55], and VizWiz-Captions [56] have been investigated and concluded that only MSCOCO is suitable for the evaluation of our proposed approach. Flickr offers two types of sub-datasets named Flickr8k and Flickr30k. Flickr8k includes 8000 images consisting of 6000 training, 1000 test and 1000 validation images. In Flickr30k, there are 29783 training, 1000 test, 1000 validation and a total of 31783 images. On the other hand, the VizWiz dataset includes 23431 training, 8000 test and 7750 validation images captured by people who are blind. The MSCOCO dataset has more than 120,000 images along with 5 captions for each image. This dataset dominates the image captioning studies with its grammatically and semantically correct captions with many diverse images which lead us to choose the MSCOCO dataset.

To analyze the performance of the compared approaches, several metrics including BLEU-n, ROUGE-L, and CIDEr are employed. BLEU-n compares a machine-generated caption by n-gram pairs to the human-generated ground truth captions [41]. Searching for pairs on short captions may arise a problem as getting a higher score, even the result is not correct. Hence, BLEU-n applies the brevity penalty to overcome the problem. ROUGE-L determines the longest co-occurring in word-sequence n-grams by itself [44]. CIDEr metric calculates the resemblance of the generated caption to a series of ground truth captions [43]. These metric results highly correlate with human judgments. The higher results are obtained on these metrics, the better captions are produced by the image caption generator. Among these three metrics, CIDEr is the only one designed for the image captioning problem while others are originally derived for machine translation. Therefore, methods are sorted based on CIDEr metric in the next.

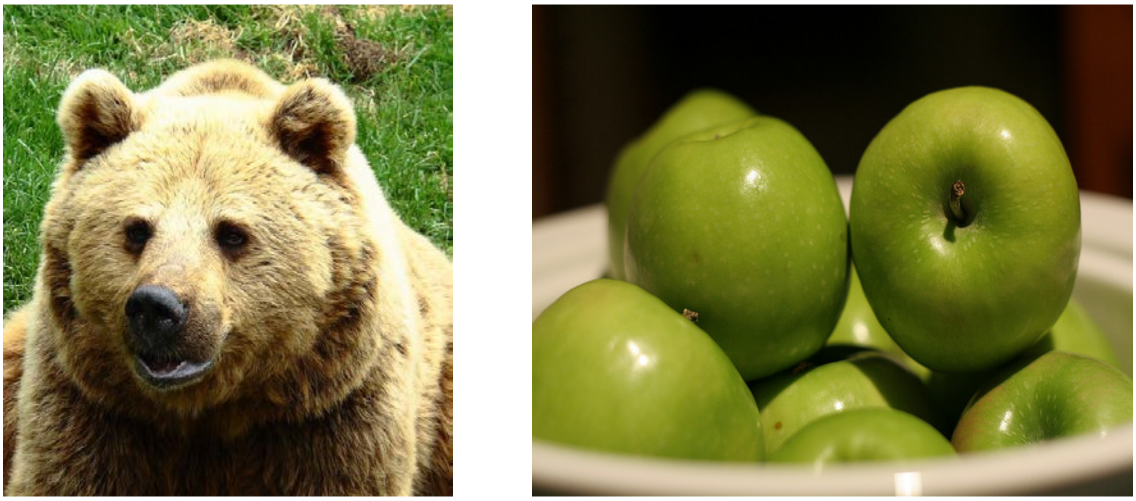
In the experiment with the decoder part, the RMSprop optimizer was employed to update the parameters. The learning rate was set to be $1x10^{-3}$. The loss function was chosen as cross-entropy loss with the combination of the negative log likelihood loss and the logarithmic softmax function. The size of the feature vector was 2048. The vocabulary size, embedding size and hidden size of GRU layers were set to be 10.000, 128 and 512, respectively. The input size of the FC layer is equal to the hidden size of GRU whereas the output size is the vocabulary size. An activation function *linear* was employed in the FC layer.

4.2 Results and Discussion

The proposed deep GRU based decoder firstly was tested with three different encoders in order to find the best CNN architecture compatible with our multi-layer GRU design. The Inception-v3, Xception, and NASNetLarge were evaluated under BLEU-n, ROUGE-L, and CIDEr metrics, and the scores were given in Table 1 which demonstrates that the proposed captioning approach is consistently better with the Inception-v3 CNN encoder.

Table 1 Performance of Deep GRU-Based Decoder with Three Different CNN Encoders

Encoders	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**ROGUE-L**	**CIDEr**
NASNetLarge	0.644	0.455	0.312	0.213	0.473	0.748
Xception	0.650	0.466	0.324	0.225	0.479	0.770
Inception-v3	**0.652**	**0.470**	**0.330**	**0.232**	**0.484**	**0.775**



(a) (b)

Figure 4 Sample Images from MSCOCO Dataset

To demonstrate the generated captions, two images were selected from MSCOCO dataset given in Figure 4 while the ground truth and generated captions were given in Table 2. If the generated captions with Inception-v3, Xception and NASNetLarge are compared, it can be seen that the caption from Inception-v3 is semantically more meaningful and closer to the ground truth captions.

Table 2 Ground Truth and Generated Captions for the Images of Figure 4

Ground Truth Captions for Figure 4 (a):	**Ground Truth Captions for Figure 4 (b):**
A big burly grizzly bear is show with grass in the background.	A large white bowl of many green apples.
The large brown bear has a black nose.	A white bowl of green granny smith apples.
Closeup of a brown bear sitting in a grassy area.	A white bowl filled with green Granny Smith apples.
A large bear that is sitting on grass.	A bowl filled with many shiny green apples.
A close up picture of a brown bear's face.	A bowl full of fresh green apples are kept.
Generated captions:	**Generated captions:**
Inception-v3: **a grizzly bear is laying down on a green field.**	**Inception-v3**: **a pile of green apples on a table.**
Xception: a large brown bear laying on top of a tree.	Xception: a bowl of fruit sitting on a table.
NASNetLarge: a grizzly bear sitting on a rock with a grassy field behind him.	NASNetLarge: a plate topped with a green apple and a green apple.

The Inception-v3 based image caption generator has been integrated into the *VirtualEye+* as its outstanding score compared to the other CNN encoders. The caption generation time is about 10 seconds depending on the internet connection for the smartphone application. The proposed approach is also compared with those of Chen et al. [27], You et al. [28], Xu et al. [31] and Mathews et al. [31] under available metrics reported on their papers. In each column, the highest score is indicated with bold fonts and the approaches are sorted based on the CIDEr metric. Even though the results of [31] are slightly better than ours in terms of the BLEUn metrics, the proposed approach outperforms the others with respect to ROGUE-L and CIDEr metrics.

Table 3 Performance Metric Results

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROGUE-L	CIDEr
[27]	0.505	0.308	0.191	0.121	-	0.600
[28]	0.510	0.322	0.207	0.136	0.390	0.654
[31]	**0.664**	**0.482**	**0.337**	0.233	**0.484**	0.689
[29]	0.653	-	-	**0.238**	-	0.769
Proposed Approach	0.652	0.470	0.330	0.232	**0.484**	**0.775**

5. Conclusion

Herein, we proposed a new image captioning approach based on the Inception-v3 CNN encoder and deep GRU decoder. The deep decoder design has been investigated in natural language expressions of images with multi-layer sequential GRUs. The proposed approach was trained on the MSCOCO dataset. The results showed that captioning performance was significantly improved with a multi-layer GRU based decoder and outperformed the state-of-the-art approaches. Then, the proposed approach was integrated with the *VirtualEye+* Android application to offer easy-operation of image captioning under a user-friendly interface. An image can be taken either from the gallery or camera to transfer to the remote server via the Firebase. The remote server runs our proposed approach to generate captions which can be transferred back to the application either in English or any other language that the user requests. In addition, *VirtualEye+* provides narrator options to allow the user to hear the generated captions. In the proposed approach, CNN encoder and RNN decoder could be embedded in the smartphone application to reduce the caption generating time which could be interesting directions for future work.

References

[1] B. Makav and V. Kılıç, "A New Image Captioning Approach for Visually Impaired People," in *11th International Conference on Electrical and Electronics Engineering*, 2019, pp. 945-949: IEEE.

[2] B. Makav and V. Kılıç, "Smartphone-based Image Captioning for Visually and Hearing Impaired," in *11th International Conference on Electrical and Electronics Engineering*, 2019, pp. 950-953: IEEE.

[3] G. Kulkarni *et al.*, "Baby talk: Understanding and generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1601-1608.

[4] M. Mitchell *et al.*, "Midge: Generating image descriptions from computer vision detections," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 2012, pp. 747-756: Association for Computational Linguistics.

[5] D. Elliott and F. Keller, "Image description using visual dependency representations," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1292-1302.

[6] X. Zhang, X. Wang, X. Tang, H. Zhou, and C. Li, "Description generation for remote sensing images using attribute attention mechanism," *Remote Sensing*, vol. 11, no. 6, p. 612, 2019.

[7] H. Fang *et al.*, "From captions to visual concepts and back," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1473-1482.

[8] R. Mason and E. Charniak, "Nonparametric method for data-driven image captioning," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2014, pp. 592-598.

[9] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi, "Treetalk: Composition and compression of trees for image descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 351-362, 2014.

[10] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *Transactions of the Association for Computational Linguistics,* vol. 2, pp. 207-218, 2014.

[11] M. Yang *et al.*, "An Ensemble of Generation-and Retrieval-Based Image Captioning With Dual Generator Generative Adversarial Network," *IEEE Transactions on Image Processing,* vol. 29, pp. 9627-9640, 2020.

[12] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-rnn)," *arXiv preprint arXiv:1412.6632,* pp. 1-17, 2015.

[13] A. Oluwasammi *et al.*, "Features to Text: A Comprehensive Survey of Deep Learning on Semantic Segmentation and Image Captioning," *Complexity,* vol. 2021, 2021.

[14] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625-2634.

[15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:.* 2014.

[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104-3112.

[17] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv preprint arXiv:.* 2014.

[18] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks Learning Systems,* 2020.

[19] S. Kalra and A. Leekha, "Survey of convolutional neural networks for image captioning," *Journal of Information Optimization Sciences,* vol. 41, no. 1, pp. 239-260, 2020.

[20] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2018, pp. 8697-8710.

[21] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2017, pp. 1251-1258.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818-2826.

[23] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Computing Surveys,* vol. 51, no. 6, pp. 1-36, 2019.

[24] H. Wang, Y. Zhang, and X. Yu, "An Overview of Image Caption Generation Methods," *Computational Intelligence Neuroscience,* vol. 2020, 2020.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:.* 2014.

[27] T. Chen *et al.*, "``Factual"or``Emotional": Stylized Image Captioning with Adaptive Learning and Attention," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 519-535.

[28] Q. You, H. Jin, and J. Luo, "Image captioning at will: A versatile scheme for effectively injecting sentiments into image descriptions," *arXiv preprint arXiv:.10121,* 2018.

[29] A. Mathews, L. Xie, and X. He, "SemStyle: Learning to Generate Stylised Image Captions Using Unaligned Text," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2018, pp. 8591-8600.

[30] M. Tanti, A. Gatt, and K. P. Camilleri, "Where to put the image in an image caption generator," *Natural Language Engineering,* vol. 24, no. 3, pp. 467-489, 2018.

[31] N. Xu, A.-A. Liu, J. Liu, W. Nie, and Y. Su, "Scene graph captioner: Image captioning based on structural visual representation," *Journal of Visual Communication Image Representation,* vol. 58, pp. 477-485, 2019.

[32] O. Nina and A. Rodriguez, "Simplified LSTM unit and search space probability exploration for image description," in *2015 10th International Conference on Information, Communications and Signal Processing (ICICS)*, 2015, pp. 1-5: IEEE.

[33] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156-3164.

[34] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7008-7024.

[35] J. Krause, J. Johnson, R. Krishna, and L. Fei-Fei, "A hierarchical approach for generating descriptive image paragraphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 317-325.

[36] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4894-4902.

[37] J. Mao, X. Wei, Y. Yang, J. Wang, Z. Huang, and A. L. Yuille, "Learning like a child: Fast novel visual concept learning from sentence descriptions of images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2533-2541.

[38] T. A. Praveen and J. A. A. Jothi, "Enhancing Image Caption Quality with Pre-post Image Injections," in *Advances in Machine Learning and Computational Intelligence*: Springer, 2021, pp. 805-812.

[39] H. Wang, H. Wang, and K. Xu, "Evolutionary Recurrent Neural Network for Image Captioning," *Neurocomputing*, 2020.

[40] Y. Tao, X. Wang, R.-V. Sánchez, S. Yang, and Y. Bai, "Spur gear fault diagnosis using a multilayer gated recurrent unit approach with vibration signal," *IEEE Access*, vol. 7, pp. 56880-56889, 2019.

[41] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 311-318.

[42] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65-72.

[43] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4566-4575.

[44] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, 2004, pp. 74-81.

[45] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An Audio Captioning Dataset," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 736-740: IEEE.

[46] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Journal of Wiley Interdisciplinary Reviews: Data Mining Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[47] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[48] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 4520-4524: IEEE.

[49] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[50] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness Knowledge-Based Systems*, vol. 6, no. 02, pp. 107-116, 1998.

[51] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[52] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310-1318: PMLR.

[53] Ö. Çaylı, B. Makav, V. Kılıç, and A. Onan, "Mobile Application Based Automatic Caption Generation for Visually Impaired," in *International Conference on Intelligent and Fuzzy Systems,* 2020, pp. 1532-1539: Springer.

[54] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014, pp. 740-755: Springer.

[55] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2641-2649.

[56] D. Gurari, Y. Zhao, M. Zhang, and N. Bhattacharya, "Captioning Images Taken by People Who Are Blind," *arXiv preprint arXiv:.08565,* 2020.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 4, NO. 2, AUGUST 2021

Research Article

Experimental Analysis of Energy Efficient and QoS Aware Objective Functions for RPL Algorithm in IoT Networks

Ferhat Arat[1], Sercan Demirci[2]
[1]Department of Software Engineering, Samsun University, Turkey; ferhat.arat@samsun.edu.tr;
[2]Corresponding Author; Department of Computer Engineering, Ondokuz Mayıs University, Turkey;
sercan.demirci@omu.edu.tr; +90 362 312 19 19

Received 5 February 2021; Revised 27 May 2021; Accepted 11 June 2021; Published online 31 August 2021

Abstract

The Internet of Things (IoT) refers to smart devices with limited resources that connect to the Internet and transmit data. Routing is an important process in this structure, which can be described as the general frame of wireless sensor networks (WSNs). The Routing Protocol for Low-Power and Lossy Networks (RPL) is recommended by the Internet Engineering Task Force (IETF) to provide communication in resource-constrained networks and is designed for routing in IoT. Basically, it is the Internet Protocol Version 6 (IPv6) protocol developed based on the energy consumed by devices. The algorithm has an important place in the performance of the IoT network. In this paper, the performance of the RPL under different objective functions (OFs) is examined. OFs are symbolized and defined by detailed equations. This study provides an experimental analysis of the RPL algorithm. An overview of the RPL algorithm is also included. Finally, the RPL algorithm is simulated by a custom simulator which is performing on the application layer, created using the Python programming language. The algorithm's behaviour in terms of different OFs such as throughput maximization, energy efficiency maximization and energy consumption minimization was observed and the results were evaluated under different parameters such as packet size, number of nodes and different signal-to-noise ratio (SNR) values. Our experimental results may be useful for both researchers and practitioners working in related fields.

Keywords: Internet of Things, Wireless Sensor Network, Routing, RPL

1. Introduction

The IoT refers to the non-centralized structure created by devices that can connect to the Internet, and communicate with each others by exchanging data. In another definition [1], IoT is the technology of 'things' that transmits by collecting information over a wireless internet network. This characterization was first suggested by Kevin Ashton in 1999 [2]. The IoT structure represents an innovative technology that allows everyday objects to connect to the internet [3]-[4] and can be found in places such as smart homes, smart factories, the health industry and other systems.

The IoT includes sub-domains in terms of its scope, device, and all functions. The wireless sensor network (WSN) can be considered an IoT substructure. The WSN can be defined as networks that consist of small and cheap nodes with sensing, processing, and communication capabilities which are designed for complex monitoring operations in the geographical areas [5].

According to USNIC data, devices that work under the IoT frame will increase over the years [6] and will be used in every point of daily life to provide ease and advantage. Devices will transfer large amounts of data in terms of their transactions. However, these tiny devices have low power, lossy, and limited resources. Some devices perform under resource-constrained conditions in IoT applications [7]. Parameters such as power and throughput are instances of these limited resources.

Routing is a significant research field for wireless sensor networks as well as for standard networks. Path selection of data or processed information by specified constraints and metrics defines routing [8]. The determination of the transmission path is directly or indirectly related to various parameters. As previously mentioned, limited capacities and abilities of IoT/WSN had led to suggestions of different methods in routing. In this context, the IPv6 routing protocol RPL is proposed for resource constrained devices by the IETF working group Routing Over for Low Power and Lossy Network (ROLL) [9]. The

To Cite This Article: F. Arat, S. Demirci, "Experimental Analysis of Energy Efficient and QoS Aware Objective Functions for RPL Algorithm in IoT Networks," Sakarya University Journal of Computer and Information Sciences, vol. 4, no. 2, pp. 192–203, 2021. doi: 10.35377/saucis.04.02.874860

RPL algorithm aims to increase delivery performance in the network by providing a routing path for devices which have low power and limited resources.

In this work, we perform the incentive analysis of the RPL algorithm which is proposed in the literature. To the best of our knowledge, no experimental analysis work which is comparing routing path performance as throughput, energy consumption, and energy efficiency metrics. Many experimental studies compare the RPL algorithm with different algorithms in terms of different metrics. In this study, performance outputs of the routing path created by the RPL is observed simultaneously in terms of the mentioned metrics. We proposed different scenarios with various parameters such as packet size, SNR values. We are interested in evaluating the RPL performance and metrics to propose the most efficient OF to choose the best path to destination. In this study, analyses of energy consumption, energy efficiency, throughput behaviour, and end-to-end delay of RPL algorithm were conducted. First, the energy consumption of the nodes using different metrics and formulas was evaluated. Then the energy consumption of the overall network and energy efficiencies of nodes on generated topology was examined. The results were analysed in a Python-based network simulation environment. The rest of the paper is structured as follows: Section 2 deals with the related works of the RPL algorithm in terms of energy and other metrics; Section 3 provides an overview of the RPL and RPL basic structure; Section 4 describes OFs of RPL; Section 5 shows simulation parameters and a simulation of the algorithm; results are given in Section 6; and conclusions are provided in Section 7.

2. Related Work

The RPL is included as standard routing protocol in the literature [10]. Researchers have proposed various models to develop the standard RPL routing protocol. The green routing protocol was proposed for the Internet of Multimedia Things (IoMT) [11]. The IoMT is a more developed version of the RPL algorithm and works by choosing one parent node according to metrics such as delay or battery consumption. The proposed algorithm takes into account minimizing of energy consumption and carbon footprint. Thus, it aims to provide Quality of Service (QoS) parameters. Chang et al. [12] proposed an energy-oriented routing protocol by improving the existing routing protocol of the RPL. The Expected Transmission Count (ETX) and remaining energy metrics were combined, generating the OF. The routing path was chosen according to this generated OF. Thus, energy consumption balance was provided not only over the nodes but also over the whole network. The proposed scheme was simulated with the Cooja network simulator and compared with previous RPL metrics. In research by Iova et al. [13], an energy aware protocol was designed to minimize global energy consumption. Their protocol works on the basis that each node consumes the same energy amount to extend network lifetime. The designed protocol specifies network bottlenecks and provides energy efficient consumption on network overall. The OF was proposed by using the Expected Life Time (ELT) metric to generate an energy balanced network. The algorithm tackles instant lifetime of nodes and bottleneck lifetime cases during the parent selection. The proposed model was simulated in WSNet with the RPL algorithm. The RPL algorithm causes significant packet losses due to reasons such as route instability. In order to prevent packet losses, a novel approach was developed by Boubekeur et al. [14]. The solutions of the problems which exist in the RPL algorithm are generally based on developing of metrics and functions. A solution was developed based on the restriction of the maximum number of child nodes during the tree generation of a node to address the RPL's route instability issue. The proposed model is called the Bounded Degree RPL (BD-RPL). As the proposed model uses control messages in RPL, it works without a link quality measure. The results were obtained by Cooja network simulation in terms of energy consumption, delay, and transmission rate. Pereira et al. [15] proposed a new routing metric for the RPL. This metric gives not only provides a reliable path selection like ETX on the network, but also important results such as load balancing and lifetime increasing on the network. Their proposed method is known as the Network Interface Average Power (NIAP). Verification of the method was done by randomly distributed homogeneous topology with the same initial battery levels. According to the results, the NIAP metric provides 24% better results than ETX in terms of network lifetime and 1% increase in packet transmission rate.

Load balancing and congestion problems in RPL were tackled by Kim et al. [16]. They stated that most of the packet losses in heavy network traffic are caused by congestion and the reason for the parent selection process in the RPL. A Queue Utilization-based RPL (QU-RPL) model was proposed by improving the RPL algorithm to solve this problem. Their aim was to increase the end-to-end packet delivery performance by balancing traffic load within a routing tree. Results were shown by comparing an RPL using the OF0 objective function called Tiny RPL based on hop count. In order to solve the issue of load balancing, Lin et al. proposed a simple power control mechanism which works by specifying the transmission power of each node according to own link and queue losses [17]. The generated OF tackles energy losses over queue and links. A routing tree is created according to the traffic load on the network and packet delivery performance is increased.

Hoghooghi et al. suggested RPL developments for static and mobile Low Power and Lossy Networks (LLNs) which provide QoS guarantee [18]. Constraints in standard OF were improved by using the Objective Function based on Fuzzy Logic (OF-FL). Performance metrics such as number of hops, end-to-end delay, and ETX were considered. The OF allows the choice of the best candidate transmission method in terms of four main criterions. Besides the OF, RPL based on the Corona (CO-RPL) mechanism was also proposed with an aim to increase service quality on dynamic and static network structures. An RPL based on Fuzzy Logic method provides improvements on the best path selection. With CO-RPL, improvements were made to QoS due to mobility difficulties in WSNs.

An energy efficient routing protocol was proposed by Barbato et al. [19]. The RPL based protocol is called Resource Oriented and Energy Efficient (ROEE) and multiple routing metrics were used in the proposed model. Energy amount and battery consumption metrics were created as OFs instead of only the hop count metric. The proposed ROEE algorithm was compared with basic RPL protocol in terms of metrics such as network life time, power consumption, and the number of alive nodes. In Zhao et al. [20], a model providing energy efficiency and data transmission without sacrificing reliability was proposed and called the Energy Efficient Region-based Routing Protocol (ER-RPL). The main motivation of the work was difficulty in achieving success on parameters such as reliability and energy efficiency. Improvements were made in the downward phase called the downward route discovery in the RPL. It was proposed to use only a subset of a node rather than all nodes during route discovery. This approach has been key to providing energy efficiency.

Table 1 Classification of analyzed routing algorithms

Ref.	E2E Delay	Energy Consumption	Throughput	Energy Efficiency	PDR	Num of Hop	ETX
[11]	✓	✓	✗	✗	✗	✗	✗
[12]	✗	✓	✗	✗	✗	✗	✓
[13]	✗	✓	✗	✗	✗	✗	✓
[14]	✓	✓	✗	✗	✓	✗	✗
[15]	✗	✓	✗	✗	✗	✗	✓
[16]	✓	✗	✗	✗	✓	✗	✗
[17]	✗	✓	✗	✗	✓	✗	✗
[18]	✓	✗	✗	✗	✗	✓	✓
[19]	✗	✓	✗	✗	✗	✓	✓
[20]	✗	✓	✗	✓	✗	✗	✗

E2E Delay: End-to-End Delay **PDR:** Packet Delivery Ratio **ETX:** Expected Transmission Count

In Table 1, classification of analyzed routing algorithms is presented, and furthermore intended success measurements are shown. Each routing study investigate the RPL algorithm in terms of specified metrics.

3. RPL Overview

The RPL is a 6LoWPAN protocol designed for LLNs. The protocol was proposed by the IETF Roll working group and is classified as a distance vector and source algorithm in terms of its working principles and usage. It is designed in IEEE 802.15.4 PHY and MAC layers [21]. The routing path and routing tables are updated periodically because nodes propagate broadcast messages to identify changes in the network topology. Thus, the algorithm is classified as a proactive routing algorithm. Data transmission flows such as multipoint-to-point, point-to-multipoint and point-to-point are supported by the RPL algorithm [22]. The RPL algorithm generates the network topology as a Directed Acyclic Graph (DAG). The data transmission path is optimized by a DAG root which acts as a root node, or in other words, a sink. An example of a root node is a gateway that acts as the network's exit point to the Internet. A Destination Oriented Directed Acyclic Graph (DODAG) is a multiple DAG structure which is combined of DAG trees. It consists of multiple parent nodes in contrast to standard tree structure. DODAG structure also consists of multiple paths from leaf nodes to roots [11], which is the main difference of DODAG from the standard tree structure. Figure 1 illustrates an instance of an RPL tree.

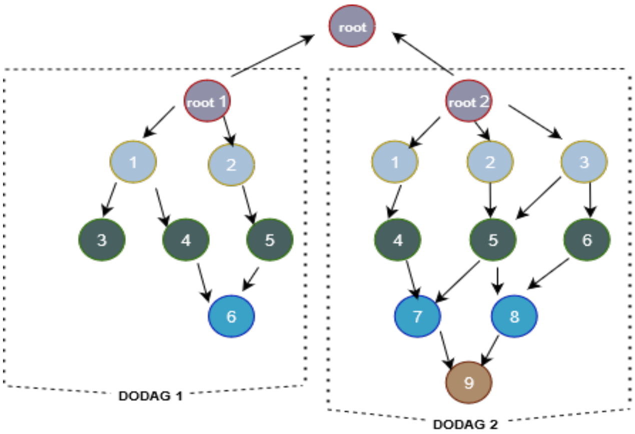


Figure 1 An RPL tree instance consisting of multiple DAGs

Tree structures consisting of nodes use identifiers for each object they contain. The RPL Instance ID is the ID of the RPL to which the DIO message is sent and the DODAG ID is the identifier number of the topology section where the RPL is sent. The DODAG Version Number and Rank provide values for nodes.

The RPL algorithm consists of two stages: downward and upward transmission. Downward transmission provides point-to-multipoint (P2MP) and point-to-point (P2P) communication, whereas upward transmission creates point-to-point (P2P) traffic. Before these stages, it is necessary to establish the network, message transmission, and to generate the routing path by using control packages. Thus, the RPL uses tree basic ICMPv6 packet types [23].

DODAG Information Objects (DIO), are the messages sent from root node to child nodes during the downward transmission phase. Figure 2 illustrates the DIO message flow.

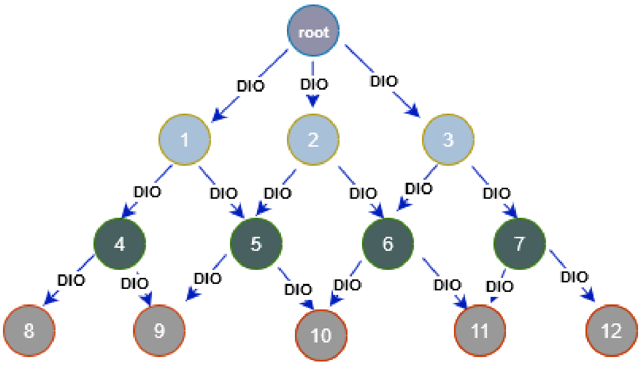


Figure 2 DIO message flow

Destination Advertisement Object (DAO) are the messages sent from nodes which have no child node to root nodes during the upward transmission phase. Figure 3 illustrates this flow.

If a new node wants to join the network, it uses multicast broadcasting through the DODAG Information Solicitation (DIS) message type and chooses a possible parent according to the status of the network.

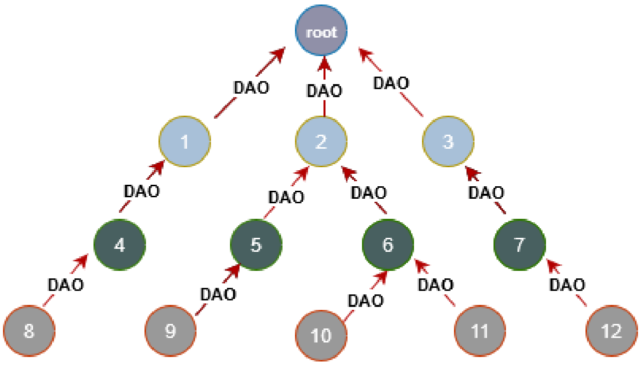


Figure 3 DAO message flow

Each node has a rank value according to its distance from the root. The rank is used for specifying the position of nodes according to roots. This value increases the further away from the root node a particular node is. The root has zero rank value on the tree. Leaf or child nodes have greater rank value than parent nodes. The rank value is a significant part of the algorithm in DODAG to prevent loops.

The algorithm starts to perform using downward transmission. The downward phase starts from the root node and ends with the discovery process on the network by DIO messages that are spread by the root node to all child nodes. DIO messages are transmitted using the Trickle Timer (RFC6206) [24]. A trickle timer is used for preventing inconsistencies that may occur due to message transmissions between nodes. In the upward stage, after spreading DIO messages, the parent selection process is conducted according to desired metrics and constraints. Each node chooses its own parent according to a calculated OF during DIO spreading. OF is necessary for parent selection of child nodes. For instance, the expected result may be minimum delay, energy consumption, or maximum message delivery rate. OF provides the desired routing path by calculating according to specified constraints and rank values. Parent selection is managed by DAO messages.

The RPL algorithm is classified according to a storage shape that is established after down and upward message transmissions and parent selection. With this perspective, the algorithm works in two basic modes: non-storing mode or storing mode. In non-storing mode, routing information is not kept by intermediate nodes but rather is stored by the DODAG root node. Route information from source to destination is determined via the root node. Packets always arrive over the root node meaning that there is no memory consumption for each node. In storing mode, each node stores route and parent information about other sub-nodes. The routing path is kept by all nodes. Path information formed during non-storing and storing modes and a general RPL view are illustrated in Figure 4.

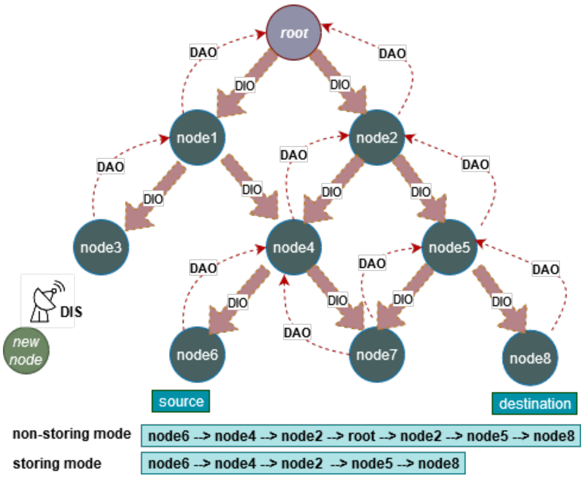

root
DAO
DIO
node1
node2
node3
node4
node5
node6
node7
node8
DIS
new node
source
destination
non-storing mode
node6 --> node4 --> node2 --> root --> node2 --> node5 --> node8
storing mode
node6 --> node4 --> node2 --> node5 --> node8

Figure 4 RPL general view and modes

4. Objective Functions

The RPL routing protocol provides routing metrics via OFs. These functions are determined by desired routing acts such as the minimum distance path selection, the minimum energy consumption and delay, and maximum packet delivery ratio. Developments in the RPL also differ in behaviour of these functions. OF which is mainly categorized as the link and node metrics is decisive on network performance [25].

The RPL algorithm chooses the routing path according to various OFs. Link and communication quality are specified by the OF. In this work, the network model is represented by DODAG *G = (V, E)*, where *V* is a set of IoT device nodes and *E* is a set of all communication links in the graph *G*. In addition, various OFs were used as routing metrics and are summarized in Table 2.

4.1 Throughput Maximization (P1)

In the scope of work, the throughput is represented by bandwidth and, capacity. It is a link and communication parameter between two nodes. This objective function P1, given in Equation 1, is used to find links that have maximized their capacity. Throughput should be maximized for network performance. The throughput can be represented as *B* and calculated according to the following formula:

$$B = (max) \sum_{\forall l_{i,j} \in P} \delta_{ij}\, W \log_2(1 + SNR_{i,j}\,)\ \ (bps) \tag{1}$$

where the *P* represents the set of all paths on the topology and where $l_{i,j}$ is a link over the path P.

$$\delta_{\mathrm{ij}} = \begin{cases} 1, & \text{if there is a positive data flow over the link } \mathrm{l_{i,j}} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where the δ_{ij} represents the data flow over the link, i.e., $\delta_{ij} = 1$ if there is a flow over the link and it is considered while calculating the energy consumption, $\delta_{ij} = 0$ otherwise. *W* represents bandwidth in Hertz. In this work, *W* is considered constant and $SNR_{\mathrm{i,j}}$ is the signal-to-noise-ratio of link $l_{i,j}$ assigned from randomly selected different mean values.

4.2 Energy Consumption Minimization (P2)

Energy Consumption is the total amount of energy or power that nodes on the network spend for operations on the network. This objective function P2 as described in Equation 3 is used to find nodes that consume energy at the minimum level. Energy consumption should be minimized for all nodes. It is represented as EC and defined as follows:

$$\mathrm{E_C} = (\min) \sum_{\forall \mathrm{l_{i,j}} \in \mathrm{P}} \delta_{\mathrm{ij}} \left(\mathrm{ETX} + \ \mathrm{E_{Circuitry}} + \mathrm{E_{Idle}}\right) + \sum_{\forall \mathrm{l_{i,j}} \notin \mathrm{P}} \delta_{\mathrm{ij}}\, \mathrm{E_{Idle}} \quad (\text{joule}) \tag{3}$$

where the *ETX* is the Expected Transmission Count by RFC6551. It is frequently used in LLNs. ETX is the successful transmission of the packet from the source node to the destination and can be defined as follows:

$$\mathrm{ETX} = \mathrm{Pt_{t_x}} \qquad (\text{Joule}) \tag{4}$$

$$\mathrm{t_{t_x}} = \frac{\text{packet size}}{\mathrm{B}} \quad (\text{sec}) \tag{5}$$

where the *P* in the definition is power and considered a constant.

$E_{Circuitry}$ is the energy consumption power of devices due to their individual characteristics and can be defined as follows:

$$\mathrm{E_{Circuitry}} = \mathrm{P_C t_{t_x}} \quad \text{(Joule)} \tag{6}$$

where the P_C is the device power value and considered a constant.

E_{Idle} is the energy consumed by devices when they do not perform any operation on the network and is defined as follows:

$$\mathrm{T_{remain}} = \mathrm{T} - \mathrm{t_{t_x}} \quad \text{(sec)} \tag{7}$$

$$\mathrm{E_{Idle}} = \mathrm{P_{Idle} T_{remain}} \quad \text{(Joule)} \tag{8}$$

where the P_{Idle} is the power spending value of node when it is not operating.

4.3 Energy Efficiency Maximization (P3)

Energy Efficiency is defined as the output received against the energy consumed per node over a period of time under consideration [26]. It is the metric that determines the energy consumption on the network. This objective function P3, provided by Equation 9, is used to find maximum energy efficient nodes on the network. Maximizing the energy efficiency of the nodes ensures that energy consumption on the routing is minimized. Energy Efficiency is represented by E_E and defined as follows:

$$\mathrm{E_E} = (\max) \ \frac{\mathrm{R}}{\mathrm{E_C}} \quad \left({}^{\text{bits}}/_{\text{joule}} \right) \tag{9}$$

$$\mathrm{R} = \mathrm{B} \times \mathrm{T_s} \quad \text{(bits)} \tag{10}$$

where the R is the maximum number of bits that can be carried over the link.

4.4 End-to-End Delay Minimization (P4)

End-to-End delay refers to the time it takes for a packet to be transmitted from source to destination over the network. It also can be calculated as the sum of propagation delay, transmission delay, queuing delay, and processing delay.

Table 2 Metric notations

Symbol	Definition
B	Channel throughput
ETX	Expected transmission count
t_{t_x}	Rate of packet number that is sent by channel throughput
E_C	Energy consumption for network overall
$E_{Circuitry}$	Energy consumption power of devices due to individual characteristics
E_{Idle}	Total energy consumption amount of the device when it is idling
P_{idle}	Idle power value of devices
P_C	Characteristic power value of devices
T_{remain}	Remaining time value in the time slot of each device after processing
E_E	Energy efficiency of each node
R	*M*aximum number of bits

5. Simulation Environments

In this section, the simulation parameters and results which evaluate the performance of the RPL algorithm are presented through Python programming language. The generated custom simulator performs on application layer. To analyse the general behaviour of the RPL under different conditions, various simulation settings were considered. The symbols and simulation parameters are shown in Table 2 and Table 3. These simulations are based on a static DODAG network topology consisting of a single root. Nodes are distributed randomly on network topology and the transmission distance between nodes is assumed as a node transmits message only to their neighbours for all scenarios.

The algorithm was simulated for different scenarios. On the constant node number scenario, 20 randomly distributed nodes were generated for network topology. The DODAG Id value was never used since the model has a single root node. Message class was generated in three basic message types: DIO, DAO and DIS. As stated in section 3.1, the identity information of each node and the RPL tree were considered. In the remaining scenarios, the number of nodes were set at 5, 7, 9, 10, 13 and 20 and all were considered static. An iteration limit of 20 was set for each different parameter and packet time interval was 0.5 seconds. SNR values used in these simulation were separated by different means. The SNR value of a link was assumed to follow an exponential process with different means and these values were generated randomly by various means. Each node had the same time slot.

Table 3 Simulation parameters

Symbol	**Definition**	**Values**
t	Simulation Time	20 iterations
L	Packet Size	[100,1000] bytes
I	Packet Interval	0,5 sec
n	Number of Nodes	[5, 20]
snr	SNR value means	[2.5, 20] dB
P	Power	1.98 Watt
W	Channel Bandwidth	$5*10^6$ Hertz
P_c	Circuitry power	0.21 Watt
P_{idle}	Idle power	0.99 Watt
T_s	Time slot	100 sec

Evaluation of the protocol was performed using performance metrics such as energy consumption, average network throughput, efficient path throughput and average end-to-end delay. Table 4 outlines the principal characteristics of the RPL simulation.

Table 4 Principal Characteristics of the RPL Simulation

Abbreviation	**Definition**
Algorithm	RPL
Simulation Tool	Python
Topology	Random distributed, fixed
Message Flows	MP2P, P2P, P2MP
Control Messages	DIO, DAO, DIS
Metrics	OF and rank
Mode	Non-Storing
QoS Aims	Energy consumption, energy efficiency, throughput, end-to-end delay, hop count

6. Results

This section presents the simulation results which evaluate the performance of the RPL algorithm by using Python. The simulation results are shown as the average value of different parameters and functions. The simulation time was set at 20 iterations. As shown in Table 3, the bandwidth was set to 5×10^6 Hz. Other parameters were considered constant values. Before the simulation started, source and destination addresses were specified and given as input. For each iteration, different performance metrics were calculated.

Figure 5 displays the comparison of throughput of the RPL algorithm against various SNR values ranging from 2.5 to 20 dB, and packet size is assumed to be 500 bytes. It was observed that if the algorithm was aiming for throughput maximization, the energy consumption and energy efficiency metrics acted as non-optimal. The reason for this behaviour is that the algorithm selects parents only by considering throughput without the other metrics. Also, since an increase in the SNR value also increased the signal quality, it positively affected the throughput value.

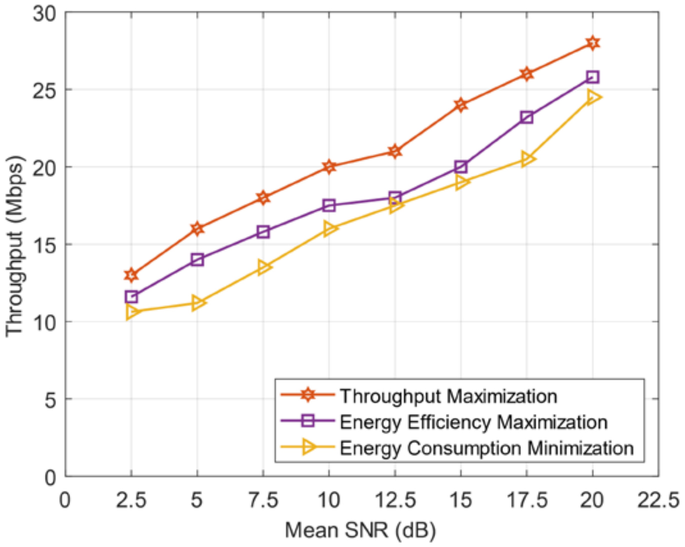


Figure 5 Throughput of selected routing paths for different SNR values under various OFs

Figure 6 shows the throughput of selected routing path for 200, 300, 400, 600, 800 and 1000-bytes packet sizes respectively. The SNR value for running this simulation was considered at 10 dB. The algorithm selected the routing path according to all three metrics. In addition, when the packet size increased, the throughput value decreased. Increasing of packet size reduced the capacity value, but the capacity optimal OF provided the highest capacity according to other objective functions.

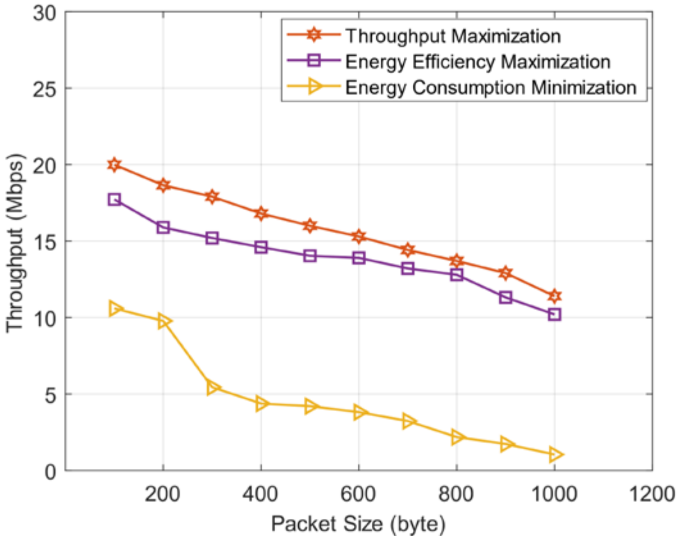


Figure 6 Throughput of selected routing paths for different packet size values under various OFs

Figure 7 illustrates energy consumption of routing paths under other throughput and energy efficiency metrics. As seen in the figure, when the SNR value increased, throughput, energy efficiency and energy consumption also increased. The algorithm selected the routing path and parent nodes according to minimum energy consumption. Thus, the energy consumption value of the path was smaller than the other metrics which were considered.

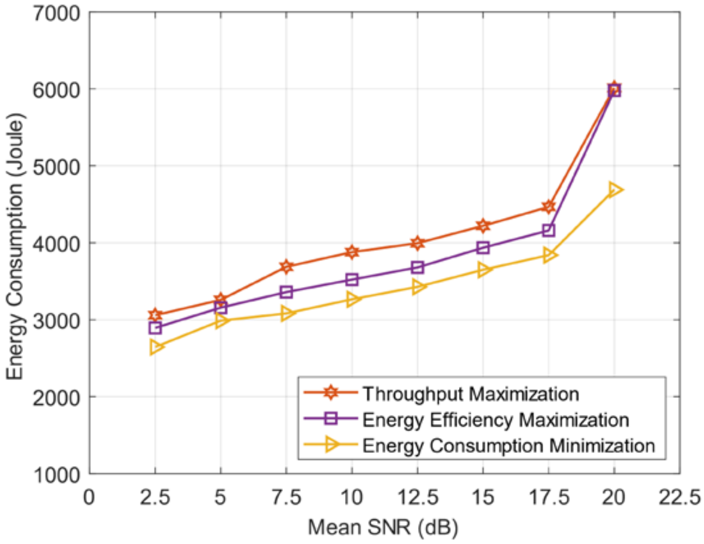


Figure 7 Energy consumption of selected routing paths for different SNR values under various OFs

Figure 8 demonstrates the energy efficiency of nodes over the routing path. While generating the routing path, nodes choose the most energy efficient node as the parent node. The higher the SNR value, the higher the communication quality and transmission performance. Accordingly, the number of bits sent per unit energy increased.

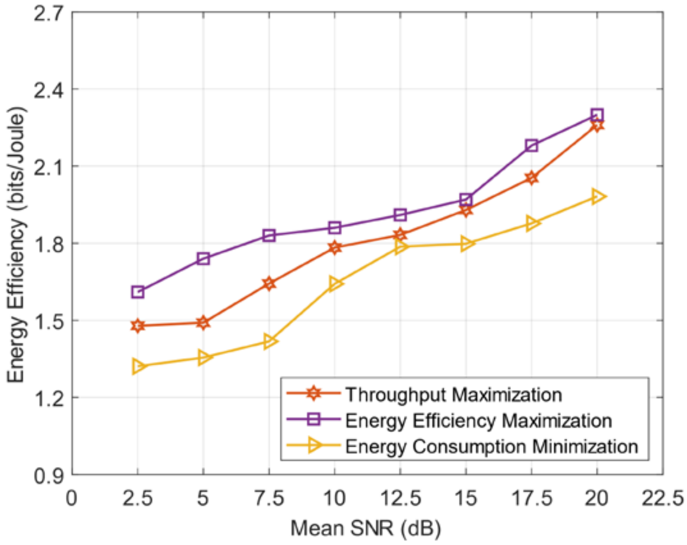


Figure 8 Energy efficiency of selected routing paths for different SNR values under various OFs

Figure 9 shows the comparison of end-to-end delay of any randomly selected routing paths in the RPL tree for 5, 7, 9, 10, 13, 15, 17, and 19 node counts, respectively. From source to destination, end-to-end delay increased when the node number increased. As the number of nodes increased, the number of unit transactions on the network increased and the queuing, processing, transmission, and propagation delay increased accordingly.

7. Conclusion

The RPL is a significant routing protocol used in low resource networks. It determines the routing path in the communication and data exchange of devices in the network in accordance with intended performance metrics. The RPL protocol, which works in both storing and non-storing mode, can be

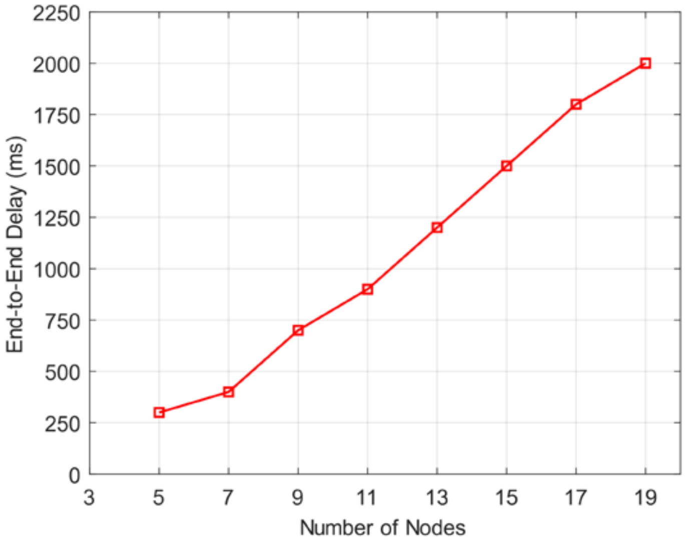


Figure 9 End-to-end delay of selected routing paths for different numbers of nodes in the RPL tree

developed in many ways. Since the root node stores all routing path information in non-storing mode, it creates an exponential storage of path information in memory, which can increase memory cost. In this article, an overview of the RPL algorithm which works on IPv6 networks was provided and the parent selection principles were explained. OFs used during parent selection were also defined. The algorithm was simulated using the Python programming language and by simulating different OFs via various parameters such as packet size, SNR values, and number of nodes. The RPL algorithm behaviour was observed by throughput maximization, energy efficiency maximization, energy consumption minimization, and end-to-end delay OFs and results were evaluated.

References

[1] I. M. Shehabat and N. Al-Hussein, “Deploying internet of things in healthcare: Benefits, requirements, challenges and applications,” *J. Commun.*, pp. 574–580, 2018.

[2] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, & M. Dohler. Standardized protocol stack for the internet of (important) things. *IEEE Communications Surveys Tutorials*, 15(3):1389–1406, 2013.

[3] N. Benamar, A. Jara, L. Ladid, and D. E. Ouadghiri, “Challenges of the internet of things: IPv6 and network management,” *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 328–333, 2014.

[4] E. Borgia, “The Internet of Things vision: Key features, applications and open issues,” *Comput. Commun.*, vol. 54, pp. 1–31, 2014.

[5] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Comput. netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[6] L. Atzori, A. Iera, and G. Morabito. Th e internet of things: A survey. *Computer Networks*, 54(15):2787 – 2805, 2010.

[7] Y. Shin and S. Seol, “Improvement of power consumption in RPL-based networks for mobility environment,” *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–3, 2020.

[8] F. Arat and S. Demirci, “Energy and QoS aware analysis and classification of routing protocols for IoT and WSN,” *2020 7th International Conference on Electrical and Electronics Engineering (ICEEE)*, pp. 221–225, 2020.

[9] T. Winter, et al. ,RPL: IPv6 routing protocol for low-power and lossy networks, *RFC 6550* (2012) 1–157, 2012.

[10] M. A. Mahmud, A. Abdelgawad, and K. Yelamarthi, “Improved RPL for IoT Applications,” *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 988–991,2018.

[11] S. A. Alvi, G. A. Shah, and W. Mahmood, “Energy efficient green routing protocol for Internet of Multimedia Things,” *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pp. 1–6, 2015.

[12] L.-H. Chang, T.-H. Lee, S.-J. Chen, and C.-Y. Liao, “Energy-efficient oriented routing algorithm in wireless sensor networks,” *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3813–3818, 2013.

[13] O. Iova, F. Theoleyre, and T. Noel, “Using multiparent routing in RPL to increase the stability and the lifetime of the network,” *Ad Hoc Netw.*, vol. 29, pp. 45–62, 2015.

[14] F. Boubekeur, L. Blin, R. Leone, and P. Medagliani, “Bounding Degrees on RPL,” *Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, 2015.

[15] H. Pereira, G. L. Moritz, R. D. Souza, A. Munaretto, and M. Fonseca, “Increased network lifetime and load balancing based on network interface average power metric for RPL,” *IEEE Access*, vol. 8, pp. 48686–48696, 2020.

[16] H.-S. Kim, H. Kim, J. Paek, and S. Bahk, “Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks,” *IEEE Trans. Mob. Comput.*, vol. 16, no. 4, pp. 964–979, 2017.

[17] M. Lin, H.-S. Kim, and S. Bahk, “Transmission power control for large scale industrial applications in low power and lossy networks,” *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 380–382, 2015.

[18] S. Hoghooghi and R. Javidan, “Proposing a new method for improving RPL to support mobility in the Internet of things,” *IET netw.*, vol. 9, no. 2, pp. 48–55, 2020.

[19] A. Barbato, M. Barrano, A. Capone, and N. Figiani, “Resource oriented and energy efficient routing protocol for IPv6 wireless sensor networks,” *2013 IEEE Online Conference on Green Communications (OnlineGreenComm)*, pp. 163–168, 2013.

[20] M. Zhao, I. W.-H. Ho, and P. H. J. Chong, “An energy-efficient region-based RPL routing protocol for low-power and lossy networks,” *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1319–1333, 2016.

[21] O. Gaddour, A. Koubâa, and M. Abid, “Quality-of-service aware routing for static and mobile IPv6-based low-power and lossy sensor networks using RPL,” *Ad Hoc Netw.*, vol. 33, pp. 233–256, 2015.

[22] C. Lim, “A survey on congestion control for RPL-based wireless sensor networks,” *Sensors (Basel)*, vol. 19, no. 11, p. 2567, 2019.

[23] A. J. H. Witwit and A. K. Idrees, “A comprehensive review for RPL routing protocol in low power and lossy networks,” *Communications in Computer and Information Science, Cham: Springer International Publishing*, pp. 50–66, 2018.

[24] Levis, P., Clausen, T., Hui, J., Gnawali, O., & Ko, J. The trickle algorithm (rfc 6206). *Internet Engineering Task Force (IETF)*, 1-13, 2011.

[25] H. Lamaazi and N. Benamar, “A comprehensive survey on enhancements and limitations of the RPL protocol: A focus on the objective function,” *Ad Hoc Netw.*, vol. 96, no. 102001, p. 102001, 2020.

[26] S. Bayhan and F. Alagoz, “Scheduling in centralized cognitive radio networks for energy efficiency,” *IEEE Trans. Veh. Technol.*, vol. 62, no. 2, pp. 582–595, 2013.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 4, NO. 2, AUGUST 2021

Research Article

Blockchain-Based Secure Credit Card Storage System for E-Commerce

Ahmet Ali Süzen[1], Burhan Duman[2]
[1]Corresponding Author; Isparta University of Applied Sciences, Turkey;
ahmetsuzen@isparta.edu.tr; +90 246 214 6581
[2]Affiliation; Isparta University of Applied Sciences, Turkey; burhanduman@isparta.edu.tr;

Received 12 March 2021; Revised 31 May 2021; Accepted 14 June 2021; Published online 31 August 2021

Abstract

Recently, serious damages have occurred in e-commerce applications due to rapidly increasing data leaks and end-user vulnerabilities. Although the source of the vulnerabilities is different, attacks result in the theft of unsafe data. In particular, the theft of credit card information reveals a financial loss. In this study, a blockchain-based secure storage model has been developed in order to prevent the theft of credit card information in e-commerce applications as a result of a possible data leak. In the sample e-commerce application developed with ASP.NET, data other than credit cards are stored. Credit card data is transmitted to the blockchain over the API with SSL protection in the e-commerce application. The blockchain model was developed using MongoDB with the BigchainDB framework. The data in each block of the blockchain is encrypted with Advanced Encryption Standard (AES) 256 bits. The data integrity of the block is provided by the SHA256 algorithm. it is aimed to protect credit card data from a possible data leak with the proposed BigchainDB-based blockchain model.

Keywords: blockchain, encryption, e-commerce, secure payment

1. Introduction

In e-commerce applications, in the payment step of the shopping made by the users; Credit cards, Money Orders / EFT, and cash on delivery options are preferred [1]. The preferred payment method is stored in application databases due to its ease of use. With increasing cyber-attacks recently, data leaks are occurring in e-commerce applications [2]. As a result of these data leaks, unauthorized transactions are made using the credit card information of the customers and material losses occur. These problems arising from the storage of customer credit cards pose a threat to both customers and e-commerce applications.

E-commerce applications are responsible for the secure storage of credit card information in the database. Although the use of SSL / TLS in client-server communication of e-commerce applications provides communication security, it does not provide database security [3]. At this point, data encryption methods are used to ensure security. The security of credit cards is tried to be determined by the Payment Card Industry-Data Security Standard (PCI-DSS) standard, recommended by Mastercard and VISA [4]. PCI-DSS is the common security standard for the use, protection, storage, and transmission of credit card data. Although the PCI-DSS standard is mandatory in e-commerce applications, potential vulnerabilities may arise due to configuration errors, developer errors, or technical errors. Possible problems encountered in e-commerce applications are listed as follows [5].

- Using weak encryption algorithms
- Encrypting data with weak or short keys
- SSL/TLS certificate authority is not preferred
- Hosting the database and the application on the same network
- Lack of updates in database and software
- Lack of vulnerability detection or non-repetition

The most difficult part of storing sensitive data is ensuring data security and integrity. For this, Data Loss Prevention (DLP), intrusion detection (IDS), or prevention systems (IPS) are widely used [6].

To Cite This Article: A. A. Suzen, B. Duman, “Blockchain-Based Secure Credit Card Storage System for E-Commerce,” Sakarya University Journal of Computer and Information Sciences, vol. 4, no. 2, pp. 204–215, 2021. doi: 10.35377/saucis.04.02.895764

Current solutions are based on policy and rule-making principles. In other words, security weaknesses occur in possible wrong or incomplete configurations [7]. Recently, blockchain technology using cryptographic methods has been introduced to protect data security and integrity [8].

Blockchain technology emerged in 2008 with a study by Satoshi Nakamoto named "Bitcoin: Peer-to-Peer Electronic Cash Payment System" [9] and the structure that forms the basis of Bitcoin, which is presented as a crypto digital currency. In order to eliminate technology centralization, it stores copies of data by distributing copies of data to users on the network, using strong encryption methods, and based on consensus. Although it has been associated with the financial sector since it was first introduced with Bitcoin, the increasing and widespread studies show that the blockchain is a security-purpose database that can be used in different sectors.

End-user grievance resulting from data leaks and weaknesses in e-commerce applications is the motivation for the study. In this study, a blockchain-based model is presented for the secure storage of credit card information in e-commerce applications. SHA256 hash algorithm is used for the data integrity of the model. Since the hash is a one-way function, encrypted data cannot be restored. Therefore, the credit card information reused by the customer is encrypted with the AES symmetric algorithm. In order to test the proposed system, an e-commerce application was developed with ASP NET and the data transmitted to the blockchain with API. In the last step, 50 different credit card information was added to the blockchain and temporal performance measurements were carried out.

2. Related Work

Blockchain technology has wide use in cryptocurrency (finance), health, insurance, logistics, advertising, copyright protection, energy, and social applications [10-11]. Although it is not a very easy technique that can be applied in every field, application trials and developments continue. Although the services of the blockchain were used from the first digital currency to smart contracts, the security of this technology is based on cryptography [12].

There are problems such as fraud, commission fees, the payment between buyer and seller, and unauthorized use of personal data in E-commerce systems that have developed with digitalization. Blockchain technology has the potential to offer reliability and transparency with payments and smart contracts. [13-14]. The use of blockchain in cyberattack prevention will be able to gain gains by preventing the damage, loss, and abuse caused by security vulnerabilities. In addition, huge damages to institutions, individuals, or governments in monetary terms due to cyber-attacks can be avoided by the application of this technology [15-16].

In their studies Shaikh and Iliev presented a transaction processing system that provides secure transactions in E-commerce and a model that protects E-commerce transactions against denial of service (DoS) attack. [17]. The transaction processing system is designed using blockchain technology, zero-knowledge proof, and modified elliptic curve cryptography encryption. The Transaction processing system designed has increased the security of general E-commerce transactions by providing privacy and integrity services. With these two security solution models offered, it is easier to protect the confidentiality and integrity of E-commerce transactions.

In cross-border e-commerce and supply chain management using blockchain technology, models and methods framework have been developed in which the key recovery problem can be successfully solved, protection against clone attack, fake tag attack, and fake product attack. The framework includes a number of blockchain-based models, including a multi-chain model, data management model, and block structure model [18].

In a study investigating the resistance to cyber-attack types in distributed systems, a blockchain-based communication architecture was proposed to guarantee the integrity assurance and permanent recording of the messages exchanged between all parties, including Unmanned Aerial Vehicles and ground control stations in the military autonomous system network. The proposed architecture has been shown to protect against data integrity compromise and authentication spoofing attacks [19].

In line with the studies in the literature, security, data integrity, transparency and performance increase stand out. Most of the studies were conceptual and theoretical, and some were experimentally applied. Although blockchain technology has been emphasized in terms of providing security, there are still a variety of security problems. Research on blockchain security is mostly technical; important business, organizational and operational problems were overlooked [20-22]. Some security gaps can also be seen in the Bitcoin structure. There is a wallet structure that a person needs for blockchain-based Bitcoin, and although it is safer to use the extended public key in this structure, the wallet can still be compromised. The extended public key can reveal the chain code, which has an important role in deriving the key, and the attacker can brute force attack all chain codes using the public key together with a public chain code [23].

As the literature studies are examined, the use of cryptocurrencies in payment methods and their security are oriented due to the increasing number of cryptocurrencies with blockchain technology. Unused studies with blockchain technology focus on secure communication, not secure storage of credit card information in e-commerce applications. These studies ensure security by using different encryption algorithms in the communication of services or applications. Although it is generally advocated not to store credit card information, many applications store information. Credit card leaks, which have increased recently, prove this. In the proposed model, it appears that it provides aspects of privacy, integrity, non-repudiation, and auditability in both communication and storage of credit card information (Table 1). The study aims to securely store credit card information in e-commerce applications without the need for storage techniques belonging to a different source or institution.

Table 1. Comparison of studies on the storage of credit card information

Studies	Blockchain	Storage Location	Encryption	Privacy	Integrity	Non-repudiation	Auditability
Our Model	+	Cloud	AES 256	+	+	+	+
[24]	-	No	RSA	+	+	-	-
[25]	-	No	RSA	+	+	-	-
[26]	-	Cloud	Scale-based Secure sensitive data (SSSD)	+	-	-	+
[27]	-	No	Hybrid the El Gamal encryption scheme with RSA	+	+	+	-
[28]	-	No	Secure Online Transaction Algorithm	+	+	-	-

3. Blockchain

Blockchain is also known as a distributed ledger in which all digital transactions between the participants, called nodes, are recorded. Unlike a centralized system that needs verification by a single authority, the blockchain offers a distributed system that performs decentralized authentication in which different nodes in its network communicate with each other using the peer-to-peer protocol (Figure 1).

The blockchain consists of linked list-like block sequences that store information such as the timestamp and transaction data along with the encrypted hash value (function) of the previous block in its own network. The Hash value of the previous block is written to the next block, a connection is established between the two blocks and a chain of consecutive blocks is formed (Figure 2). This process, which is repeated in a sequence, verifies the integrity of the previous block up to the first block known as the genesis block. The hash value of each block represents a unique code that belongs to that block and is derived from the records it contains using the SHA-256 algorithm. If the information in the block is changed, the hash value of the block also changes. Accordingly, data in any block cannot be changed retrospectively without changing the hash values of all subsequent blocks.

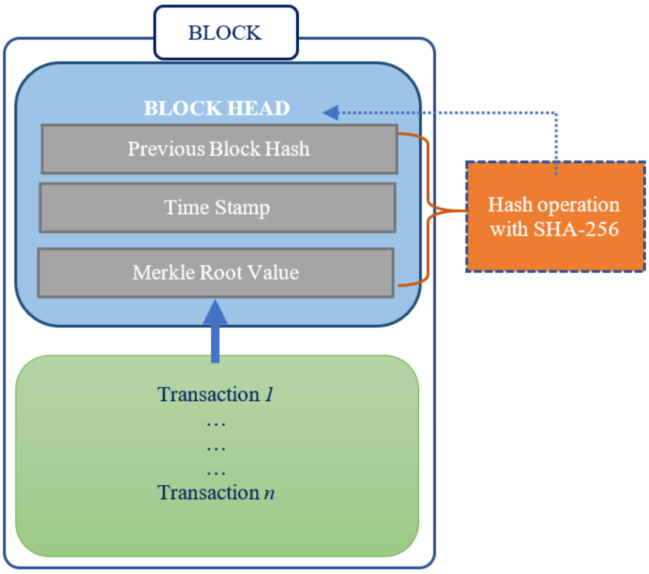


Figure 1 Block Structure

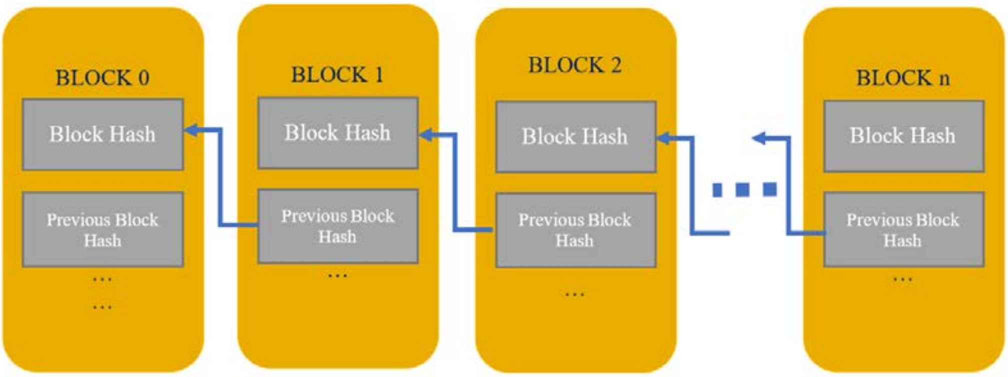


Figure 2 Basic Blockchain Representation

Commonly, Blockchain is classified into two groups as public and private. Consortium Blockchain is also encountered as a third type [29].

A public blockchain is an important structure in eliminating privacy, central authority cost, and ensuring data integrity. The private blockchain stands out in providing a central authority structure, where transactions are kept under control, data are not open to the public, and transaction speed is required. Consortium blockchain can be thought of as a semi-private blockchain. It has a structure in which fewer nodes join the network compared to the general blockchain.

In the public (distributed and open) blockchain network, data storage may be inconvenient in terms of security and privacy. Although the data is encrypted and secure in the distributed blockchain structure, there may be a possibility of leaking the information of the people with the key. In the special blockchain structure, permission must be obtained from the network structure in order to access the stored data. It can be preferred in cases where private blockchain transactions are carried out in a controlled manner, the data should not be open to the public, and the system is expected to operate quickly. Using a private blockchain structure/network in cases such as information leakage can alleviate concerns [30].

In order for a transaction to be valid and occur in the blockchain network, each block needs to look at the hash value of the previous block and have the correct hash value [31]. In case of an attack on the network and an attempt to change the information of any block, the hash value associated with the block will also change. It can be understood that there is an attack on the network since the changed hash value will not match the original. Since changing the hash value with an external intervention will break the connection between the blocks, access to other blocks will be eliminated and in this sense, data security will naturally be provided.

4. Model Architecture

In the proposed study, a local private blockchain-based database model has been developed to securely store credit card information in e-commerce applications when needed. The study consists of two parts. In the first part, an e-commerce application has been developed to collect credit card data. In the second part, data from the e-commerce application is stored in the blockchain structure. Credit card information of successful payments made in the e-commerce application is stored in a block of the block chain. As can be seen in the architecture of the model proposed in Figure 5, every $β_n$ data coming from the application reaches the block chain server via API services. Here, the data is encrypted with AES and the previous hash data is added for data integrity to form the chain block. Encryption algorithms are divided into symmetric and asymmetric. There is a public key in symmetric encryption and a private key in asymmetric encryption. Distribution of private key in asymmetric encryption causes performance slowdown. In this study, the symmetric encryption algorithm AES was preferred. AES algorithm; It has been selected from hundreds of algorithms by the National Institute of Standards and Technology (NIST) based on many criteria such as robustness, fast working on hardware and software [32].

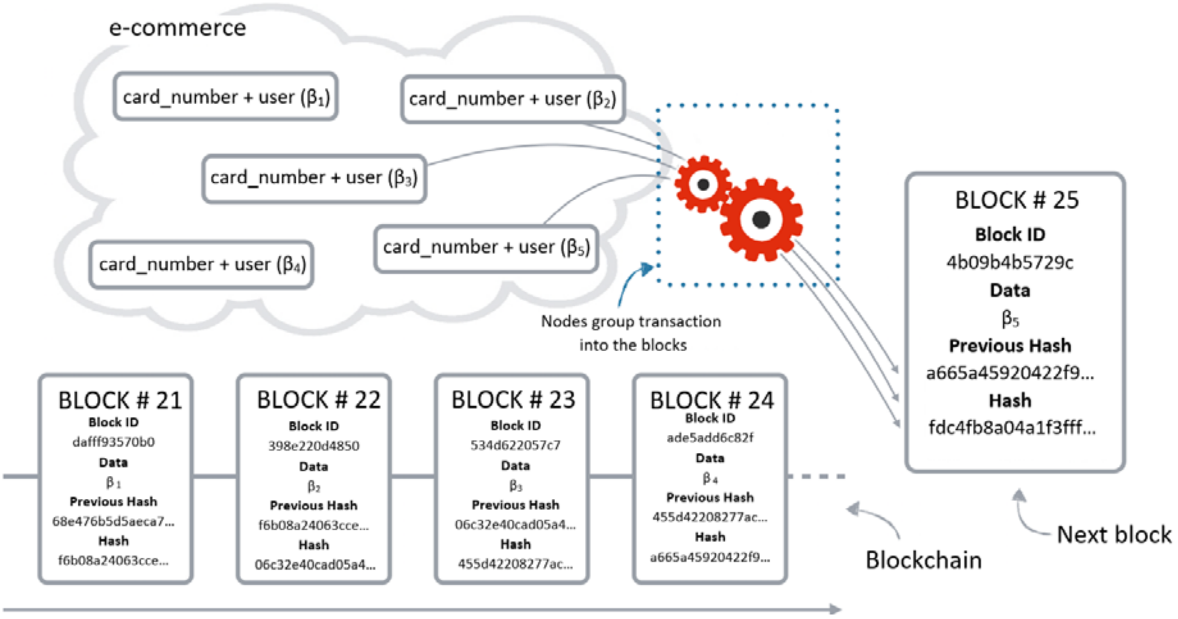


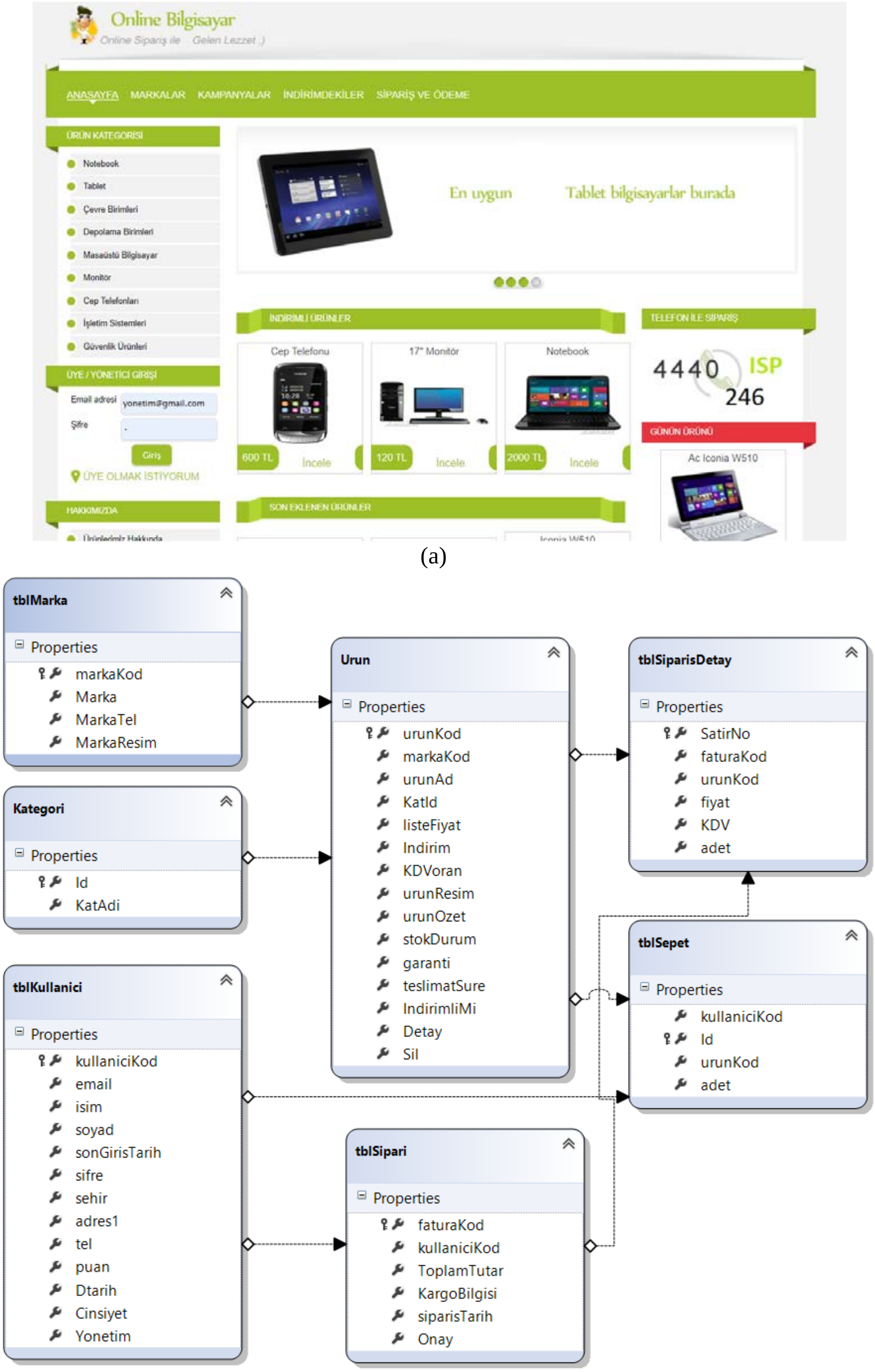
Figure 5 Working Scheme of the Proposed Blockchain Structure

4.1 Development of E-commerce Application

For testing the proposed blockchain-based security data storage system, the e-commerce application whose interface is shown in Figure 6 (a) has been developed. In the application, 3 layered architecture is used as presentation, business and data layer. In the presentation layer; A ready-made e-commerce template was placed on the ASP.NET architecture and formal (HTML + CSS) changes were made. In the data layer; The relational database shown in Figure 6 (b) has been created in the MS-SQL database. Sample computer products have been added to this database for test use. In the business layer; Database query functions are coded using the LINQ query structure.

4.2 Development of Blockchain-based Secure Storage Model

A blockchain to be used for database purposes is not sufficient to process high volumes of data. In other words, a huge amount of data cannot be stored in a block. It also lacks built-in search and indexing capabilities [33]. BigchainDB was used to store credit card information in the e-commerce application in the proposed study. BigchainDB combines the main advantages of distributed DBs and blockchains with an emphasis on scale. It enables querying of data on the MongoDB structure [34]. It was used 2 servers to develop and publish the applications. The e-commerce application was used in Windows IIS that has 1 CPU, 2 GB RAM, and for BigchainDB was used UBUNTU virtual server that has 2 CPU, 4 GB RAM.



(a)

tblMarka
Properties
- markaKod
- Marka
- MarkaTel
- MarkaResim

Kategori
Properties
- Id
- KatAdi

tblKullanici
Properties
- kullaniciKod
- email
- isim
- soyad
- sonGirisTarih
- sifre
- sehir
- adres1
- tel
- puan
- Dtarih
- Cinsiyet
- Yonetim

Urun
Properties
- urunKod
- markaKod
- urunAd
- KatId
- listeFiyat
- Indirim
- KDVoran
- urunResim
- urunOzet
- stokDurum
- garanti
- teslimatSure
- IndirimliMi
- Detay
- Sil

tblSipari
Properties
- faturaKod
- kullaniciKod
- ToplamTutar
- KargoBilgisi
- siparisTarih
- Onay

tblSiparisDetay
Properties
- SatirNo
- faturaKod
- urunKod
- fiyat
- KDV
- adet

tblSepet
Properties
- kullaniciKod
- Id
- urunKod
- adet

(b)

Figure 6 E-Commerce Application, a) Web Interface of the Application b) Relational Database Design

Credit card data coming from the e-commerce application will be transferred to BigchainDB Server over JSON services with HTTP API. For this, MongoDB and BigchainDB were configured on the local server with Ubuntu operating system (Ubuntu> = 16.04) (Algorithm 1). After configuration, the root url list can be accessed via BigchainDB Server localhost: 3352 /api /v1/ as shown in Algorithm 2.

Algorithm 1 BigchainDB local server configuration

```
$ export
STACK_REPO=bigchaindb/bigchaindb
$ export STACK_BRANCH=master
$ export TM_VERSION=0.22.8
$ export MONGO_VERSION=3.6
$ bash stack.sh
```

Algorithm 2 BigchainDB Server HTTP Client-Server API

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "api": {
    "v1": {
      "assets": "/api/v1/assets/",
      "blocks": "/api/v1/blocks/",
      "metadata": "/api/v1/metadata/",
      "outputs": "/api/v1/outputs/",
      "streams":
"ws://localhost:3352/api/v1/streams/valid_transactions",
      "transactions": "/api/v1/transactions/",
      "validators": "/api/v1/validators"
    }
  },
  "software": "BigchainDB",
  "version": "2.2.1"
}
```

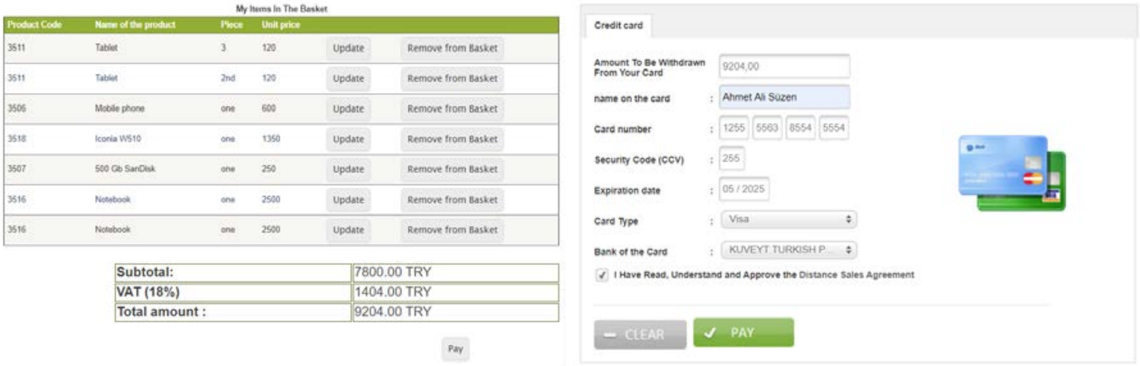
C # programming language and BigchainDB libraries are used to create the blockchain and to process incoming GET / POST requests. CREATE and TRANSFER classes are created in the block chain software and communication is carried out over these classes. The interface of block, transactions and public key requests in the blockchain software is shown in Algorithm 3.

Algorithm 3 BigchainDB Server HTTP Client-Server API

```
Get Blocks given block_id [C#]
public static async Task<Block> getBlock(int block_Id,
IBlockchainConfigurationBuilder builder = null)
Get Blocks given transaction_id [C#]
public static async Task<IList<int>>
getBlocksByTransactionIdAsync(string
transaction_Id,IBlockchainConfigurationBuilder builder = null)
Get Outputs given a public_key [C#]
public static async Task<List<OutputList>> getOutputsAsync(string
public_Key,IBlockchainConfigurationBuilder builder = null)
```

5. Testing the Application

In order to store credit card information in the blockchain structure, the user first adds the products to the basket, as shown in Figure 1, from the e-commerce application. Then he enters the credit card information (credit card information created for testing purposes) on the payment screen. After clicking the payment button, if the payment is made, the credit card information is sent to the HTTP API to be added to the block chain.



My Items In The Basket

Product Code	Name of the product	Piece	Unit price		
3511	Tablet	3	120	Update	Remove from Basket
3511	Tablet	2nd	120	Update	Remove from Basket
3506	Mobile phone	one	600	Update	Remove from Basket
3518	Iconia W510	one	1350	Update	Remove from Basket
3507	500 Gb SanDisk	one	250	Update	Remove from Basket
3516	Notebook	one	2500	Update	Remove from Basket
3516	Notebook	one	2500	Update	Remove from Basket

Subtotal:	7800.00 TRY
VAT (18%)	1404.00 TRY
Total amount :	9204.00 TRY

Credit card

Amount To Be Withdrawn From Your Card 9204,00

name on the card : Ahmet Ali Süzen

Card number : 1255 5563 8554 5554

Security Code (CCV) : 255

Expiration date : 05 / 2025

Card Type : Visa

Bank of the Card : KUVEYT TURKISH P...

☑ I Have Read, Understand and Approve the Distance Sales Agreement

Figure 7 E-Commerce application basket and payment form

Algorithm 4 Query request and transaction response into the blockchain

```
GET /api/v1/blocks/1 HTTP/1.1
Host: http://localhost:3352/
----------------------------------
HTTP/1.1 200 OK
Content-Type: application/json
{
  "height": 12,
  "transactions": [
    {
      "asset": {
        "data": {
          "b1": "554823589989898- h4k5sdkf67"
        }},
      "id":
"c5f40d3880b454c1ce659a90498c579f03173ccfdce038599d4a5c2440b30616",
      "inputs": [
        {
          "fulfillment":
"pGSAIDE5i63cn4X8T8N1sZ2mGkJD5lNRnBM4PZgI_zvzbr-cgUCy4BR6gKaYT-
tdyAGPPpknIqI4JYQQ-p2nCg3_9BfOI-15vzldhyz-j_LZVpqAlRmbTzKS-
Q5gs7ZIFaZCA_UD",
          "fulfills": null,
          "owners_before": [
            "c2FzYWRzZGRzZmRmZGZkZ2ZnZmdoZ2hnaAsdzYcWfamZsZHNqZmIgc2R"
          ]}
      ],
      "metadata": {
        "sequence": 0
      },
      "operation": "CREATE",
      "outputs": [
        {
          "data": "b1",
          "condition": {
            "details": {
              "public_key":
"4K9sWUMFwTgaDGPfdynrbxWqWS6sWmKbZoTjxLtVUibD",
              "type": "ed25519-sha-256"
            },
            "uri": "ni:///sha256;
PMIICXQIBAAKBgQCUENCfgan0HTeeHXSPCz851LxHpno43I29hZQ4LX8Ko0hXX4Zk"
          },
          "public_keys": [
"MIICXQIBAAKBgQCUENCfgan0HTeeHXSPCz851LxHpno43I29hZQ4LX8Ko0hXX4Zk"
          ]}
      ],
      "version": "2.0"
    }]}
```

The request to add the BigchainDB API to the blockchain initiates a new create transaction and saves the incoming data in the last block of the chain. The integrity of the data is ensured by the SHA 256 hash function in the block structure. For data security, each incoming data block is encrypted with AES symmetric algorithm. The *privatekey* required for decrypting the encrypted data is stored on the local server. In Algorithm 4, the display of the record made after a payment transaction and the transaction query are given.

Within the e-commerce application, the administrator can list all the data in the blockchain database if the authorized user wants it (Algorithm 5).

Algorithm 5. JSON view of data in blockchain

```
GET /api/v1/assets/?search=creditcard_db HTTP/1.1
Host: http://localhost:3352/
--------------------------------------------------
HTTP/1.1 200 OK
Content-type: application/json
[
    {
        "metadata": {"b1": "5584519788679753"},
        "id": "
7dc96f776c8423e57a2785489a3f9c43fb6e756876d6ad9a9cac4aa4e72ec193"
    },
    {
        "metadata": {"b2": "5109789159864971"},
        "id": "
4814d92093ac8a0f4a2163ab87dee509ba306a58f5888be0edcb2fcd0712028b"
    },
    {
        "metadata": {"b3": "5108451207830725"},
        "id": "
76a8277347f52530e1cf979175a178980b3a180d176165c985d85f7e142f1eed"
    }
]
```

In the proposed blockchain structure, queries are carried out over all interconnected blocks (Figure 8). Therefore, it is necessary to ensure the security of the network and web application. The internal mechanism described in the modeling of the blockchain network provides its own security. Goldfinger, Finney, Spending, Feather, Vector76, Netsplit, and Eclipse attacks are used in distributed architectures. Since a special and local blockchain structure is used with BigchainDB in this study, it will not be exposed to distributed attacks. In particular, sending data to the model via APIs may cause problems in

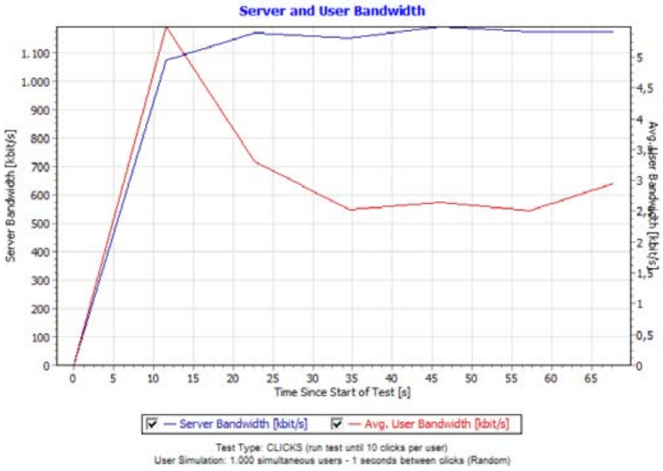


Figure 8 DDOS test attack bandwidth simulation

possible Distributed Denial of Service attacks (DDOS) attacks. Therefore, a 1000-user DDOS attack simulation was performed to see the load stress on the proposed model. According to the test results, the server can respond to what 1000 users want. Since the increase in the number of users and requests will increase the server load, there will be a delay. This problem can be overcome with application bandwidth or load balancer systems. In addition, possible attack situations can be prevented by Firewalls, IDS, or IPS systems. Although web applications do not have definitive solutions to prevent DDOS attacks, the proposed system is not considered a disadvantage.

In the second test phase, query response time was also evaluated in line with the increasing credit card data of APIs. Therefore, increasing credit card data also increases the query response time. In the BigchainDB structure, the request time account is calculated as $t_{total} = t_{in} + t_{internal} + t_{out}$ (Figure 9). Where $t_{internal}$ is the internal software query delay and it changes depending on the number of blocks in the blockchain.

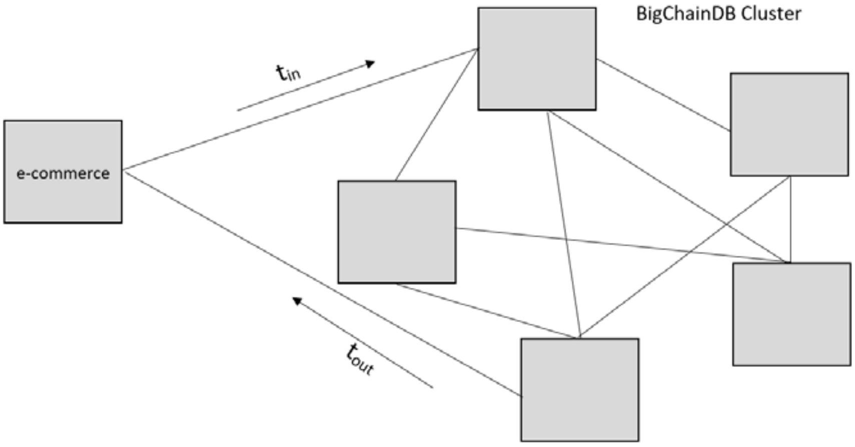


Figure 9 Query communication between BigchainDB and e-commerce application

In the e-commerce application, the total times of GET and POST requests are stored in t_{total} variable. During the test period, 50 credit card information was included in the blockchain. The times resulting from this are given in Figure 9. It is seen that the increase in the amount of data in the blockchain also increases the t_{in} and t_{out} times. In addition, the $t_{internal}$ that occurs in blockchain software creates more delay in some queries. Apache JMeter tool was used for all measurements.

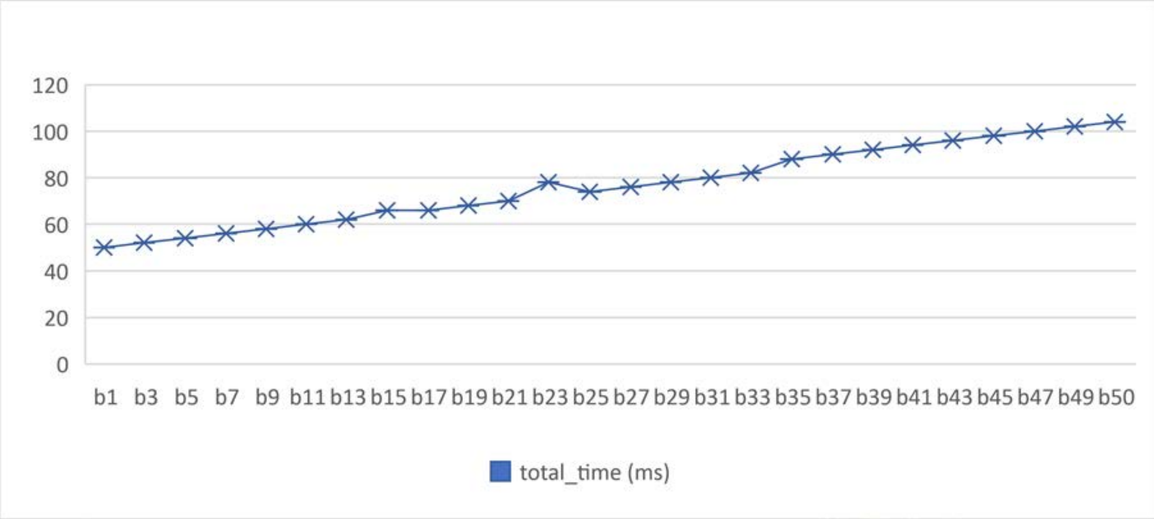


Figure 10 t_{total} Times Between BigchainDB and E-Commerce Application

6. Conclusion

E-commerce applications and the widespread use also create security weaknesses. As a result of security vulnerabilities, credit card information must be securely protected. This study, it is aimed to securely store credit card information in the blockchain-based database in an e-commerce application. An e-commerce application has been developed for the proposed model. Credit card information sent from the e-commerce application is stored in the MongoDB-based blockchain created with the BigchainDB architecture. The data in each block is encrypted with the AES asymmetric algorithm. Within the application, the authorized user can query the data with API services. When the proposed system is tested, it is seen that the blockchain structure provides data security and integrity. The increase of blocks in the blockchain structure delays the query times. This situation negatively affects the performance of the blockchain model. In future studies, it is foreseen to increase the performance by using different algorithms to encrypt and decrypt the data in the block. Also, it is aimed to develop the blockchain structure with a distributed architecture.

References

[1] M. Halaweh, “Cash on delivery (COD) as an alternative payment method for e-commerce transactions: Analysis and implications”. *International Journal of Sociotechnology and Knowledge Development (IJSKD)*, vol. 10(4), pp. 1-12,2018.

[2] S. Fatonah, A. Yulandari, and F. W. Wibowo, “A review of e-payment system in e-commerce”. *In Journal of Physics: Conference Series*, vol. 1140, p. 012033. IOP Publishing, 2018.

[3] K. F. Herkenhoff, and G., Raveendranathan” Who bears the welfare costs of monopoly? The case of the credit card industry” (No. w26604). National Bureau of Economic Research, 2020.

[4] K. Kalkan, F. Kwansa, and C. Cobanoglu, “Payment Card Industry Data Security Standards (PCI DSS) Compliance in Restaurants”. *Journal of Hospitality Financial Management*, vol. 16(2), 3, 2010.

[5] A. Ukidve, D. Smantha, and M. Tadvalka, “Analysis of payment card industry data security standard [PCI DSS] compliance by confluence of COBIT 5 framework”. *International Journal of Engineering Research and Applications*, vol.7(01), p. 42-48, 2017.

[6] W. Feng, C. Liu, Z. Guo, T. Baker, B. Cheng, and J. Chen, “Data loss prevention and storage utilization improvement of the hidden volume on mobile devices”, *In 2019 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1-6. IEEE, 2019.

[7] M. H. Furhad, S. Sadık, and M. Ahmed, “Chapter Nıne Explorıng E-Commerce In Cyber Securıty Context Through Blockchaın Technology”. *Blockchain in Data Analytics*, 2020.

[8] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, “Solutions to scalability of blockchain: A survey”. *IEEE Access*, vol. 8, 16440-16455, 2020.

[9] Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2018. [Online]. Available: https://git.dhimmel.com/bitcoin-whitepaper/. [Accessed: 21-Feb-2021].

[10] W. Chen, Z. Xu, S. Shi, Y. Zhao, and J. Zhao, “A survey of blockchain applications in different domains.” *In Proceedings of the 2018 International Conference on Blockchain Technology and Application*, pp. 17-21, 2018.

[11] M. Tekin, D. Öztürk, İ. Bahar, “Akıllı Lojistik Faaliyetlerinde Blokzincir Teknolojisi”, *Kent Akademisi*, vol. 13(3), p. 570-583, 2020.

[12] A. Ghosh, S. Gupta, A. Dua, N. Kumar, “Security of Cryptocurrencies in blockchain technology: State-of-art, challenges and future prospects” *Journal of Network and Computer Applications*, 163, 102635, 2020.

[13] X. Zhu, D. Wang, “Research on Blockchain Application for E-Commerce, Finance and Energy” *In IOP Conference Series: Earth and Environmental Science*, vol. 252, no. 4, p. 042126, IOP Publishing, 2019.

[14] L. Ismanto, H. S. Ar, A. N. Fajar, S. Bachtiar, “Blockchain as E-Commerce Platform in Indonesia”, *In Journal of Physics: Conference Series*, vol. 1179, p. 012114. IOP Publishing, 2019.

[15] S. Demirkan, I. Demirkan, A. McKee, “Blockchain technology in the future of business cyber security and accounting”, *Journal of Management Analytics*, vol. 7(2), p. 189-208, 2020.

[16] Ö. Aydın, S. Yükçü, “Siber Saldırı Önlemede Blokzinciri Teknolojisinin Fayda Maliyet Açısından Değerlendirilmesi”. *MANAS Sosyal Araştırmalar Dergisi*, vol. 9(4), p. 2519-2530, 2020.

[17] J. R. Shaikh, G. Iliev, “Blockchain based confidentiality and integrity preserving scheme for enhancing e-commerce security” *In 2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN*), pp. 155-158, 2018.

[18] Z. Liu, Z. Li, “A blockchain-based framework of cross-border e-commerce supply chain”, *International Journal of Information Management*, vol. 52, 2020.

[19] P. Angın, “Blockchain-Based Data Security in Military Autonomous Systems”. *Avrupa Bilim ve Teknoloji Dergisi*, p.362-368, 2020.

[20] W. Wang, H. Huang, L. Zhang, and C. Su, “Secure and efficient mutual authentication protocol for smart grid under Blockchain”. *Peer-to-Peer Networking and Applications*, p. 1-13, 2020.

[21] M. J. Lahkani, S. Wang, M. Urbański, M. Egorova, “Sustainable B2B E-commerce and blockchain-based supply chain finance”. *Sustainability*, vol. 12(10), p. 2-14, 2020.

[22] J. Leng, M. Zhou, L. Zhao, J. Huang, Y. Y. Bian, “Blockchain security: A survey of techniques and research directions”. IEEE Transactions on Services Computing, 2020. DOI: 10.1109/TSC.2020.3038641

[23] E. Zaghloul, T. Li, M. W. Mutka, J., Ren, “Bitcoin and blockchain: Security and privacy”. IEEE Internet of Things Journal, 7(10), 10288-10313, 2020.

[24] K. Z. Oo, “Design and Implementation of Electronic Payment Gateway for Secure Online Payment System". *Int. J. Trend Sci. Res. Dev*, vol. 3, 1329-1334,2019.

[25] P. Dijesh, S. Babu, & Y. Vijayalakshmi, “Enhancement of e-commerce security through asymmetric key algorithm”. *Computer Communications*, 153, 125-134,2020.

[26] M. Sumathi, & S. Sangeetha, “Scale-based secured sensitive data storage for banking services in cloud”. *International Journal of Electronic Business*, vol. 14(2), 171-188, 2018.

[27] J. P. Magsino, E. R. Arboleda, & R. R. Corpuz, “Enhancing Security Of El Gamal Encryption Scheme Using Rsa And Chaos Algorithm For E-Commerce Application”. *International Journal Of Scientific & Technology Research*, vol. 8(11), 2019.

[28] J. Gualdoni, A. Kurtz, I. Myzyri, Wheeler, M., & S. Rizvi, “Secure online transaction algorithm: securing online transaction using two-factor authentication”. *Procedia computer science*, 114, 93-99, 2017.

[29] M. Tanrıverdi, M. Uysal, M. T. Üstündağ, “Blokzinciri Teknolojisi Nedir? Ne Değildir? Alanyazın İncelemesi” *Bilişim Teknolojileri Dergisi*, vol. 12(3), p. 203-217,2019.

[30] Bankalar Arası Kart Merkezi, “Blockchain 101 v.2”, 2015. [Online]. Available: https://bctr.org/dokumanlar/Blockchain101v2r2.pdf. [Accessed: 05-Feb-2021].

[31] Parasozlugu,” Public (Genel) Blok Zincir (Blockchain) Nedir?” 2017. [Online]. Available: www.kriptoparasozlugu.com/genel-public-blok-zincir-blockchain-nedir/. [Accessed: 11-Feb-2021].

[32] Das, D., Danial, J., Golder, A., Modak, N., Maity, S., Chatterjee, B., Sen, S EM and Power SCA-Resilient AES-256 Through> 350× Current-Domain Signature Attenuation and Local Lower Metal Routing. *IEEE Journal of Solid-State Circuits*, 56(1), 136-150,2020.

[33] M. Simić, G. Sladić, and B. Milosavljević, “A case study IoT and blockchain powered healthcare”. *In Proc. ICET*, pp. 1-4, 2017.

[34] BlockchainDB, “Features & Use Cases”, 2018. [Online]. Available: https://www.bigchaindb.com/features/. [Accessed: 05-Feb-2021].

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 4, NO. 2, AUGUST 2021

Research Article

A Hybrid Prognostic Approach Based on Deep Learning for the Degradation Prediction of Machinery

Ahmet Kara[1]
[1]Corresponding Author Department of Industrial Engineering, Hitit University, Turkey;
ahmetkara@hitit.edu.tr

Received 09 April 2021; Revised 27 June 2021; Accepted 07 July 2021; Published online 31 August 2021

Abstract

Remaining useful life (RUL) prediction is of great significance for prognostic and health management (PHM) as it can achieve more reliable and effective maintenance strategies. With the advances in the field of deep learning, data-driven methods have provided promising prognostic prediction results. Hence, this research presents a data-driven prognostic approach based on deep learning models for predicting the RUL of mechanical systems effectively. Multiple separable convolution layers, a bidirectional Long Short-Term Memory (LSTM) layer, and fully-connected layers (FCL) are included in the proposed network, named the SC-BLSTM, to accomplish more accurate prognostic prediction from the raw degradation data acquired by different sensors. The proposed SC-BLSTM approach aims to learn complex and nonlinear features from the input data and capture temporal dependencies from the learned features. The presented approach in this research is tested and verified on the degradation data of turbofan engines (C-MAPSS dataset) from NASA. The result demonstrated that the SC-BLSTM is able to achieve more effective RUL prediction compared with some existing prognostic models.

Keywords: Remaining useful life, deep learning, separable convolutions, bidirectional LSTM, prognostics

1. Introduction

Prognostics and Health Management (PHM) is crucially significant for providing productivity and availability of modern mechanical systems. While conventional procedures like corrective maintenance and preventive maintenance have limited ability to decrease the overall cost and avoid unexpected failures, PHM provides the requirements related to more effective maintenance decision-making [1]. The fundamental objectives of PHM technology consist of avoiding unexpected system failures, minimizing the overall costs, and enhancing the reliability of the entire system. The remaining useful life (RUL) prediction of machinery has recently attracted more and more attention in PHM technologies, which is one of the most critical and challenging tasks [2].

Generally, prognostic approaches such as RUL estimation can be grouped into three categorized: model-based, data-driven, and hybrid methods [3]. Model-based techniques typically use historical degradation information to develop a mathematical model. They are less capable of predicting RUL effectively in complicated and noisy environments [4]. Conversely, data-driven methods aim to capture the degradation processing of complex mechanical systems using the past degradation data. With the recent advancements of sensor technology, data-driven models have been extensively utilized to estimate RUL effectively in the literature. Khazaee et al. [5] developed a new data-driven approach based on the artificial neural network (ANN) for RUL prediction of mechanical systems. Wang and Mamo [6] combined the gradient boosted regression (GBR) method and the artificial bee colony (ABC) algorithm to estimate the degradation progression of prismatic cells. As the last group, hybrid methods, which consist of the integration of model-based and data-driven methods, focus on taking advantage of the strengths of these models. Xue et al. [7] developed a prognostic approach that integrates unscented kalman filter and support vector regression (SVR) tuned by a genetic algorithm for RUL prediction. In this article, a novel data-driven technique is proposed for accurate and effective RUL estimation of mechanical systems.

Recently, several prognostic approaches based on deep neural networks have been extensively introduced in order to capture complex and nonlinear patterns from degradation data, and these

To Cite This Article: A. Kara, “A Hybrid Prognostic Approach Based on Deep Learning for the Degradation Prediction of Machinery,” Sakarya University Journal of Computer and Information Sciences, vol. 4, no. 2, pp. 216–226, 2021. doi: 10.35377/saucis.04.02.912154

approaches have significantly increased the efficiency and reliability of RUL prediction. Convolutional neural network (CNN) [8], long-short term memory (LSTM) [9], gated recurrent unit (GRU) [10], and some hybrid methods [11], [12] have been implemented to take advantage of their superiority. For instance, Chen et al. [13] built an encoder-decoder structure consisting of CNN, bidirectional GRU, and attention mechanism for bearing RUL prediction. Wang et al. [14] utilized multiple separable CNN and fully-connected layers (FCL) to extract high-level features from raw degradation data with the aim of estimating the RUL of machinery. A deep learning-based prognostic method with partial observations addressed by Li et al. [15] was applied to realize more accurate and effective RUL estimation using an image dataset of cutting wheel.

Li et al. [16] presented a multi-scale deep CNN structure to learn complex features from turbofan engine degradation data, and their approach provided an effective RUL prediction performance. A hybrid deep learning framework that combines LSTM network and fully-connected layer introduced by Xia et al. [17] was handled to achieve RUL prediction of mechanical systems effectively. Yang et al. [18] presented a deep learning-based approach, which is combined CNN, attention mechanism and multiple bidirectional GRU, to investigate the interpretability of the neural network. Besides, an ensemble approach including CNN and bidirectional LSTM with multiple time windows developed by Tangbin Xia et al. [19] was utilized to learn the degradation trend of machinery.

In this paper, an integrated deep learning approach, called the SC-BLSTM, including multiple separable CNNs, a bidirectional LSTM and fully-connected layers, is addressed to effectively accomplish the RUL prediction of machinery. Multiple separable CNNs with different kernel sizes are applied to obtained discriminative information from the raw degradation measurements collected with multiple sensors. The second component of the proposed network comprising a bidirectional LSTM layer is employed to capture high-level representations from the learned features. Finally, fully-connected layers are applied to estimate the RUL of machinery. Experimental results demonstrated that the SC-BLSTM approach performs better performance compared with the existing related models. The primary contributions of this research are listed as follows:

- A data-driven based on multiple separable CNNs, bidirectional LSTM, and FCL is proposed for the RUL estimation of machinery so as to automatically reveal the sophisticated and nonlinear patterns from the degradation process.
- The hyperparameters of the proposed method, including the dropout rate, the number of filters in each separable convolutional layer, the number of units and batch size, are tuned by a grid search algorithm.
- Experimental consequences on the C-MAPSS dataset indicated that the presented method is capable of providing more effective RUL prediction compared with the existing relevant studies.

The rest of this research is designed as follows. Section 2 gives details about the components of the presented approach. Section 3 explains the SC-BLSTM framework in detail. The experimental settings are introduced in Section 4. The experimental outcomes are offered to verify the effectiveness and superiority of the SC-BLSTM model. In Section 5, this paper is summarized and concluded.

2. Technical Background

In this section, it will be presented details of the background on the basic components of the network consisting of depthwise separable convolution and bidirectional LSTM.

2.1 Depthwise separable convolution

The depthwise separable convolution, which is also named separable convolution, aims to effectively extract temporal and cross-channel relationships from different sensor data. Depthwise separable convolutions have been widely applied in the field of image classification, as they reduce computational time and the number of network parameters and avoid redundant learning correlations [20]. In contrast to traditional convolution, depthwise separable convolution comprises two sections, including depthwise convolution and pointwise convolution, as seen in Figure 1. In the first step, a single convolution kernel is implemented to each input channel by depthwise convolution to capture temporal patterns separately. The number of input channels remains the same following the operation of depthwise convolution. This operation can be formulated as in Equations 1.

$$DepthConv(W_{depth}, x)_{(i,\ j)} = \sum_{h,l}^{H,L} W_{(h,\ l)} * x_{(i+h,\ j+l)} \tag{1}$$

Where W_{depth} represents the weight matrix of $DepthConv$. H and L denote the height and width of the input data. The $*$ mark is the convolution operator. (i, j) represents the coordinates of the output features. As the second step, pointwise convolutions apply a $1x1$ convolution operation by combining the outputs of the first step. This step aims to effectively extract the cross-channel patterns from different data [21]. Accordingly, pointwise convolution operation can be expressed as:

$$PointConv(W_{point}, x)_{(i,\ j)} = \sum_{k}^{K} W_k * x_{(i,\ j)} \tag{2}$$

Where W_{point} denote the weight matrix. k is the size of convolution kernel. In general, the overall operation of depthwise separable convolution can be calculated by

$$SepConv(W_{depth}, W_{point}, x)_{(i,\ j)} = PointConv(W_{point}, x)_{(i,\ j)}\ x\ (W_{point}, DepthConv(W_{depth}, x)_{(i,\ j)}) \tag{3}$$

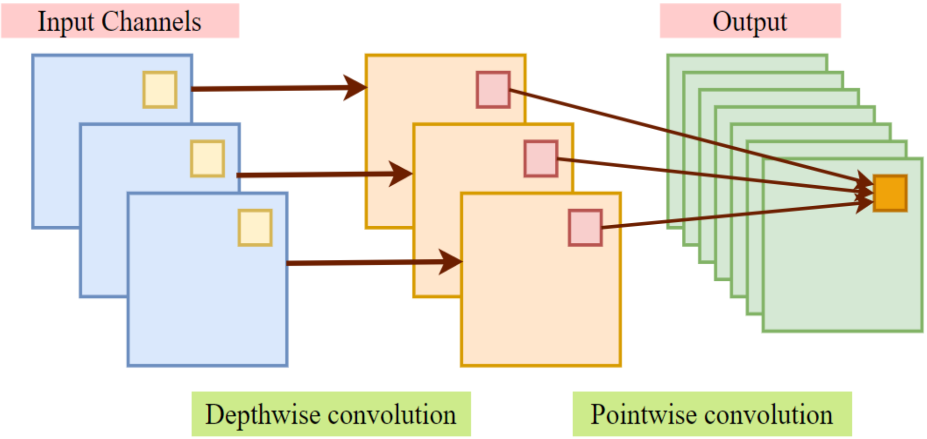


Figure 1 Depthwise separable convolution.

2.2 Bidirectional LSTM

In the SC-BLSTM model, bidirectional LSTM (BLSTM) is used to obtain temporal correlations between the features learned by separable CNN. BLSTM is widely employed in different time series prediction problems and indicates remarkable performance in RUL estimation applications [22]. The conventional LSTM layers consider the sequences of past information in a single direction. On the other hand, BLSTM consists of two different LSTM layers, which is utilized to consider past and future information. As is illustrated in Figure 2, the output of the BLSTM is obtained by combining the outputs of two layers. The overall process of BLSTM can be formulated as follows.

$$\vec{h}_t = \delta(W_{\vec{h},\vec{h}} \cdot \vec{h}_{t-1} + W_{\vec{h},x} \cdot x_t + b_{\vec{h}}) \tag{4}$$

$$h_t = \delta(W_{h,h} \cdot h_{t+1} + W_{h,x} \cdot x_t + b_h) \tag{5}$$

$$h_t = \theta(W_{h,h} \cdot h_t + W_{h,\vec{h}} \cdot \vec{h}_t) \tag{6}$$

Where $\vec{h}_t$ denotes the forward LSTM layer, h_t representes the backward LSTM layer. $\delta(\cdot)$ represents the LSTM operation. $\theta(\cdot)$ denotes the activation function.

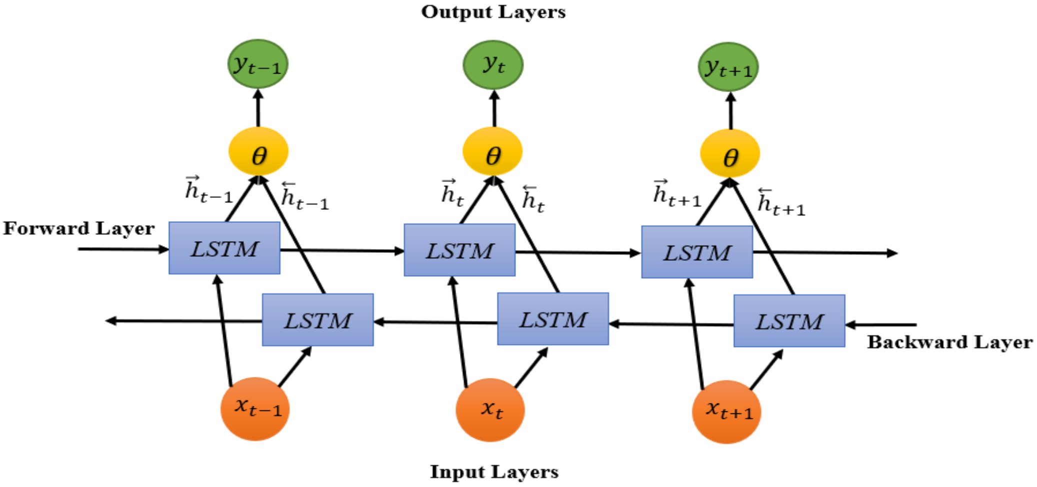


Figure 2 The Structure of Bidirectional LSTM.

3. The Proposed SC-BLSTM Model

Figure 3 illustrates the presented SC-BLSTM approach's architecture. First, two 1D separable CNN is built to automatically discover superior representations from the raw sensor measurement. The size of input data is $tw\ x\ ft$, where tw denote the time window size and ft is the number of the predetermined features. In each separable CNN, the size of the kernels is different from each other. After learning the discriminative information by two separable CNNs, a BLSTM layer is used to effectively capture temporal dependencies from the extracted features. Then, a fully-connected layer is employed to map the learned features. In the last layer, a fully-connected layer with a single neuron is utilized as the output layer of the proposed network to perform RUL prediction.

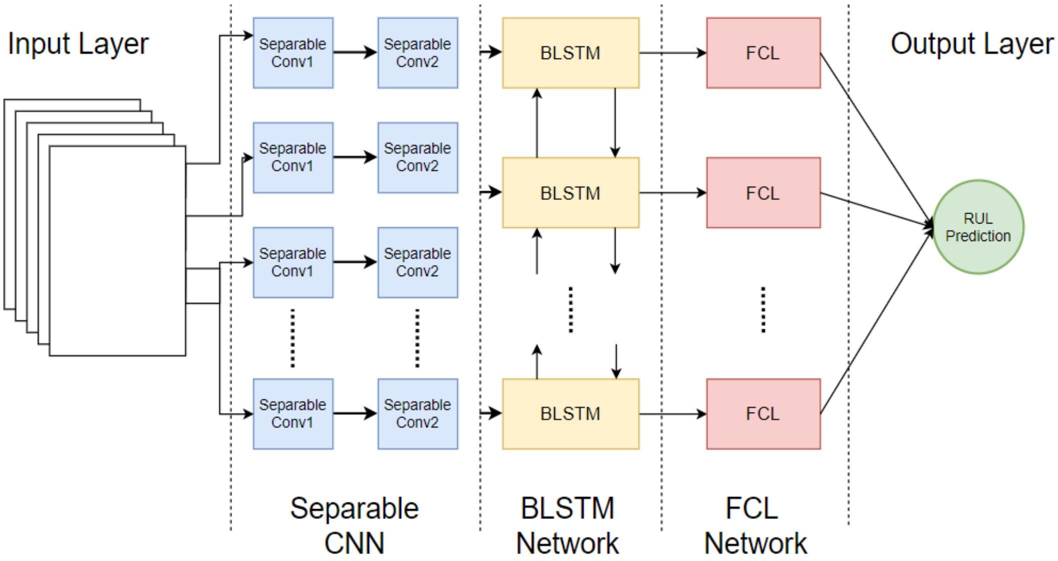


Figure 3 The Framework of the SC-BLSTM Model.

As depicted in Figure 4, the SC-BLSTM method comprises three steps; data preprocessing step, training step and testing step. In data preprocessing step, the raw degradation data is processed to perform effective prognostic prediction with the SC-BLSTM method. Training step aims to construct the optimized network using training dataset. Finally, the trained SC-BLSTM method is applied in testing step to achieve promising prognostic predictions using the testing dataset. To prevent the overfitting problem in the training step, the dropout operation is adopted after the BLSTM layer and the first FCL.

Besides, the mean square error (MSE) is applied as the loss function of the SC-BLSTM approach, which is formulated as follows.

$$MSE = \frac{1}{N_s}\sum_{i=1}^{N_s}(o_i - p_i)^2 \tag{7}$$

Where N_s is the number of the samples, o_i denotes the observed RUL values, and p_i represents the predicted RUL. In the proposed approach, the Adam algorithm is adopted to optimize the weight and biases of the network and enhance the prediction performance.

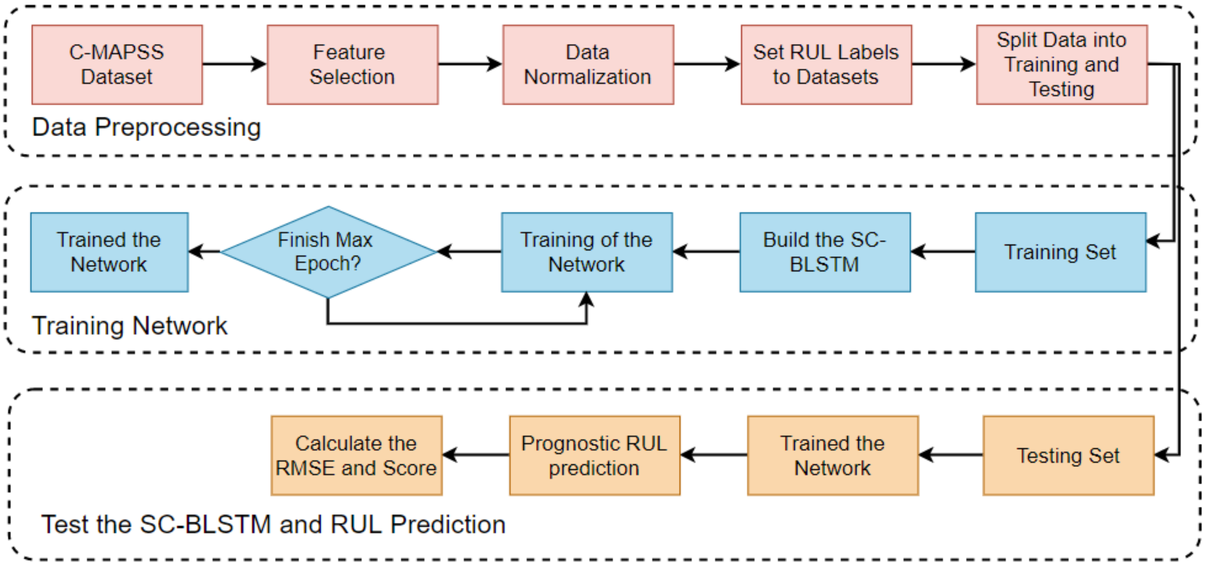


Figure 4 Implementation process of the SC-BLSTM prognostic approach.

4. Experimental Study

This section shows the RUL estimation performance of the proposed SC-BLSTM approach and comparisons with the existing works. The assessment of the prognostic prediction performance is performed on the popular turbofan engine C-MAPSS dataset.

Table 1 The details of the turbofan engine dataset

Data	**FD001**	**FD002**	**FD003**	**FD004**
Units for training	100	260	100	249
Units for testing	100	259	100	248
Operating conditions	1	6	1	6
Fault modes	1	1	2	2

4.1 C-MAPSS dataset

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset [23] introduced by NASA has been extensively utilized for the evaluations of RUL prediction. The C-MAPSS dataset is related to the aircraft turbofan engines degradation. This dataset is categorized into four subsets with varying operating conditions and fault modes, as depicted in Table 1, and each subset contains a training dataset and a testing dataset. All subsets include measurements collected by 21 sensors, engine number, time step, and three operational situations. The main aim is to predict the RUL value of each turbofan engine unit in the testing dataset. In this research, the maximum values of the training labels are clipped to no more than 125 with the aim of decreasing the prediction error when the actual RUL is greater than 125. This situation, which is a rectified RUL procedure, has been reported to remarkably enhance overall prognostic RUL prediction performance in the literature [17], [19]. The detailed definitions regarding the C-MAPSS are addressed in [23]. In this paper, FD003 and FD004 are used in experimental studies.

4.2 Experimental setting

In this article, the grid search algorithm is used to tune the hyperparameters of the proposed SC-BLSTM model for the purpose of increasing RUL estimation accuracy. The hyperparameters that need to be optimized consist of the number of filters in two separable CNN, the rate of dropout, and the number of units in BLSTM and FCL. To refrain from a large grid search space due to the computational time, the number of units and filters vary from 8, 16, 24, 32, to 48. Besides, the rate of dropout varies from 0.2, 0.3, to 0.5. Moreover, it should be noticed that 5-fold cross-validation is adopted in the training phase of each subset. The hyperparameters values tuned by the grid search algorithm and the other parameters values of the proposed network are reported in Table 2.

In the C-MAPSS dataset, several sensors have constant measurements that do not offer valuable information for RUL prediction. For this reason, these sensors measurements are removed from the input data. In this study, it is preferred 14 sensor measurements consisting of 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21 as the raw input data. Besides, the data gathered by the different sensors are standardized to be in the range $[0, 1]$ utilizing the Min-Max scaling technique by Equation 8.

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{8}$$

For the purpose of appraising the performance of the suggested SC-BLSTM method, two metrics comprising the root mean square error (RMSE) and the scoring function have been adopted in this paper. The scoring function has been widely employed by various researchers in works regarding the C-MAPSS. These metrics are calculated as follows.

$$Scoring = \{\sum_{i=1}^{N} exp\, exp\left(-\frac{d_i}{13}\right) - 1, \quad d_i < 0 \sum_{i=1}^{N} exp\, exp\left(\frac{d_i}{10}\right) - 1, \qquad d_i \geq 0 \tag{9}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2} \tag{10}$$

Where N represents the total number of the samples and $d_i = RUL_i - RUL_i'$ is the prediction error for the i data sample. In addition, the testing dataset is fed into the proposed SC-BLSTM approach trained using the training dataset to estimate the RUL of the turbofan engine, and the prediction accuracy of the SC-BLSTM model is acquired.

Table 2 Parameter settings of the experimental methods

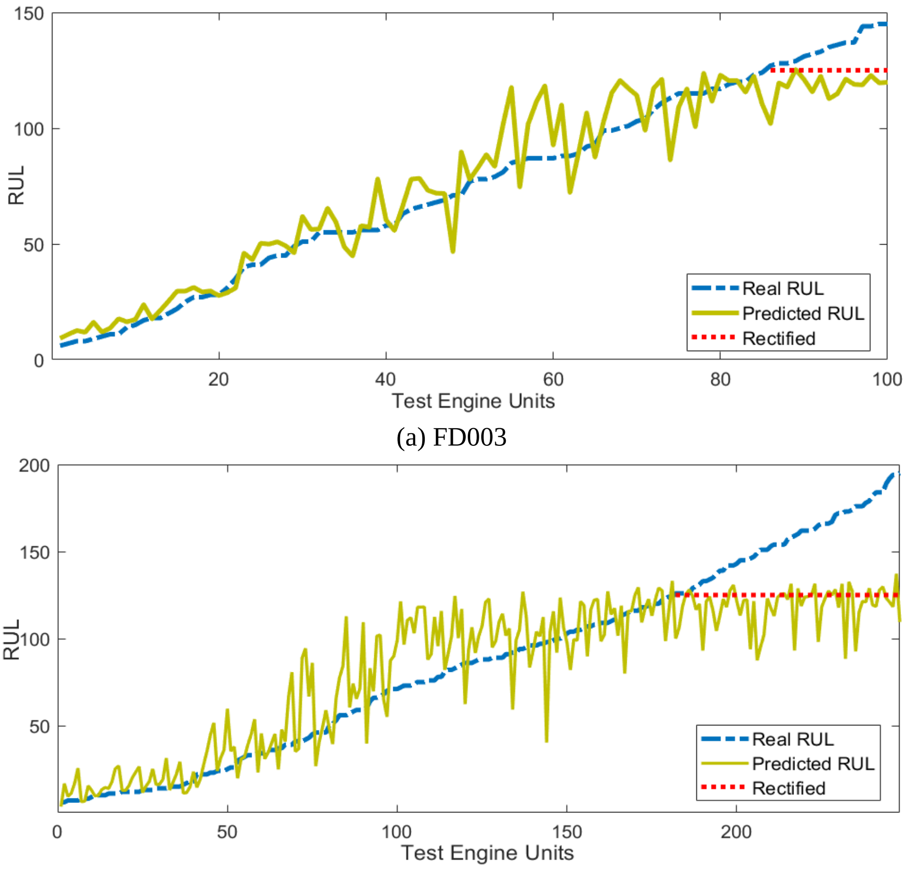
Parameter	Value (FD003 / FD004)	Parameter	Value (FD003 / FD004)
Number of filters	8-48 / 16-16	Activation function	Tanh / Tanh
Kernel sizes	3-5 / 3-5	Time window size	30 / 30
Neurons in BLSTM	48 / 8	Panding	same / same
Neurons in FCL	16/ 20	L2 regularization	1e-4 / 1e-4
Dropout rate	0.4 / 0.3	Learning rate	0.001 / 0.001
Batch size	100 / 100	R_{early}	125
Number of epochs	150 /150	Optimizer	Adam

Finally, a specific seed value is adopted to provide reproducible results from the proposed network. All experiments are conducted on a personal computer with Intel Core i7-9750H CPU, and 16 GB of RAM using the Tensorflow 2.5.0 and sci-kit learn 0.24.

4.3 Results analysis and discussions

In this section, the prognostic outcomes achieved by the SC-LSTM approach for RUL estimation utilizing the C-MAPSS dataset are analyzed and compared with the results of the existing related methods in the literature.

The RUL estimation results of the test engines according to the last recorded life cycles from the subset FD003 and FD004 are illustrated in Figure 5. The test engines are sorted by the actual RUL values for better visualization. It can be obviously observed that the prognostic prediction values of the SC-LSTM approach are capable of following the real RUL values of the testing engine units in FD003 and FD004 subsets generally. In particular, it has been seen that the prediction error tends to be more in cases where the real RUL values of the test engines are higher. Moreover, the prognostic prediction effectiveness of the presented SC-BLSTM model on the FD003 sub-dataset is superior compared with the prediction results of the FD004 sub-dataset. The reason is that the FD004 subset consists of more operational situations and fault modes. Generally, it can be observed that the presented SC-BLSTM approach has the ability to learn the degradation progression of the mechanical systems.



(a) FD003

(b) FD004

Figure 5 Comparison of prediction results for test engines sorted RUL in FD003 and FD004.

In order to investigate the proposed SC-BLSTM prediction performance for the life-time of a turbofan engine unit, the life-time prediction results of four different units in FD003 and FD004 sub-datasets are demonstrated in Figure 6. The numbers of randomly selected units from the test engines in the FD003 and FD004 sub-datasets are 39, 94, 40, and 68, respectively. It has been observed that the SC-BLSTM approach can mostly perform a remarkable RUL estimation over the degradation progression in all four examples. Furthermore, the developed SC-BLSTM approach tends to predict RUL values in the early stages to be adjacent to the rectified R_{early} as illustrated in all subplots. Afterwards, it can be said that RUL estimates tend to decline linearly until the end of their life cycle of test engine units. In the last periods of the degradation processes of 39, 94, and 40 engine units, the RUL prediction errors have a small value. This shows that the performance of the RUL prediction improves when the testing turbofan engines are close to failure. The prognostic efficiency in the last periods of the mechanical systems is important to make effective maintenance decisions, ensure system reliability and availability, and decrease the overall cost. The proposed SC-BLSTM model is able to achieve more robust and effective prognostic prediction in the last stages.

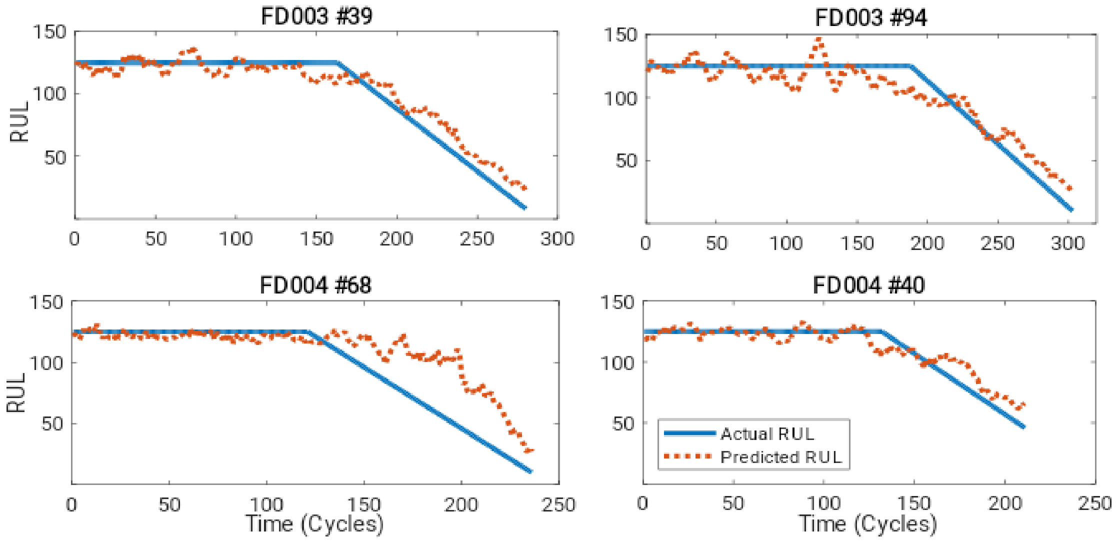


Figure 6 Examples of RUL estimations for the test engines units in FD003 and FD004.

In this research, the C-MAPSS dataset utilized to evaluate the performance of techniques developed for prognostic analysis is discussed, and the prognostic accuracy of the SC-BLSTM is compared with the existing data-driven approaches. According to RMSE metric, Table 3 reports a comparison of the SC-BLSTM with the other state-of-the-art approaches consisting of Extreme learning machine (ELM) [24], Multilayer Perceptron (MLP) [24], Support Vector Regression (SVR) [25], Relevance Vector Regression (RVR) [25], Deep belief network (DBN) [24], CNN [25], and MODBNE [24]. From Table 3, it can be said that the presented SC-LSTM approach has provided a promising prognostic accuracy compared with the results of the other benchmark approaches in terms of RMSE metric. Compared to the MODBNE approach with the second-best RMSE values on FD003 and FD004 subsets, the proposed SC-BLSTM method approximately achieves an enhancement of 1.28% and 1.40%, respectively.

Table 3 RUL prediction results of various models.

Related Approaches	**RMSE**	
	FD003	**FD004**
ELM [24]	18.90	38.43
MLP [24]	18.47	30.96
RVR [25]	22.37	34.34
SVR [25]	21.05	45.35
DBN [24]	14.71	29.88
CNN [25]	19.82	29.16
MODBNE [24]	12.51	28.66
Proposed SC-BLSTM	**12.35**	**28.26**

Table 4 illustrates the prediction performance of the SC-BLSTM method related to the scoring metric on the sub-dataset FD003 and FD004. It can be said that the prognostic prediction errors of the SC-BLSTM approach perform generally smaller results compared with the other state-of-the-art techniques in terms of the scoring metric. More specifically, the SC-BLSTM approach obtains the best prediction performance on the subset FD003. On the other hand, it performs the second-best on the subset FD004. In summary, the proposed SC-BLSTM approach has achieved a promising prognostic performance in mechanical systems taking into consideration all experimental consequences.

5. Conclusions

In this article, a deep learning-based prognostics approach, named SC-BLSTM, is addressed for the RUL prediction of mechanical systems. The presented SC-BLSTM employs the integration of the two separable CNN, a BLSTM layer, and fully-connected layers to achieve more accurate and reliable RUL

Table 4 A comparison of the Scoring metrics obtained from different approaches.

Related Approaches	**Scoring**	
	FD003	**FD004**
ELM [24]	573.78	10444.35
MLP [24]	479.85	121414.47
RVR [25]	1431.60	26509.00
SVR [25]	1598.30	371140.00
DBN [24]	442.43	7954.51
CNN [25]	1596.20	7886.40
MODBNE [24]	421.91	**6557.62**
Proposed SC-BLSTM	**222.24**	6849.18

prediction from multivariate degradation data. Firstly, two separable CNN modules are constructed to automatically learn the complex and nonlinear characteristics from the raw degradation data. Then, a BLSTM layer is leveraged to effectively capture temporal dependencies from the inputs. Finally, the extracted high-level features are fed into the fully-connected layers to achieve RUL prediction. In the SC-BLSTM approach, the dropout procedure is applied to overcome the overfitting in the training processing.

The effectiveness and superiority of the SC-BLSTM method are experimentally verified using the C-MAPSS turbofan engine dataset. The comparison between the proposed method and the existing related studies reveals that the SC-BLSTM method offers promising solutions in real-life PHM applications. More specifically, it is observed that RMSE metrics of the SC-BLSTM method on the FD003 subset have improved 6.04%, 37.69%, and 1.28%, respectively, compared with the other DBN, CNN, and MODBNE approaches. Besides, the reduction on the FD004 subset in terms of RMSE is 5.42%, 3.09%, 1.40%, respectively. It should be mentioned that although the SC-LSTM method performs the best RUL accuracy in the FD003 subset in terms of the scoring criteria, our method provides the second-best result in the FD004 subset. In future research, it will be focused on developing a hybrid structure based on transfer learning and deep learning methodologies to accomplish more effective RUL prediction of mechanical systems.

References

[1] H. Zhang, Q. Zhang, S. Shao, T. Niu, and X. Yang, “Attention-Based LSTM Network for Rotatory Machine Remaining Useful Life Prediction,” *IEEE Access*, vol. 8, pp. 132188–132199, 2020, doi: 10.1109/ACCESS.2020.3010066.

[2] B. Wang, Y. Lei, N. Li, and W. Wang, “Multiscale Convolutional Attention Network for Predicting Remaining Useful Life of Machinery,” *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7496–7504, Aug. 2021, doi: 10.1109/TIE.2020.3003649.

[3] X. Li, Q. Ding, and J.-Q. Sun, “Remaining useful life estimation in prognostics using deep convolution neural networks,” *Reliab. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018, doi: 10.1016/j.ress.2017.11.021.

[4] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, and X. Li, “Machine Remaining Useful Life Prediction via an Attention-Based Deep Learning Approach,” *IEEE Trans. Ind. Electron.*, vol. 68, no. 3, pp. 2521–2531, Mar. 2021, doi: 10.1109/TIE.2020.2972443.

[5] M. Khazaee, A. Banakar, B. Ghobadian, M. A. Mirsalim, and S. Minaei, “Remaining useful life (RUL) prediction of internal combustion engine timing belt based on vibration signals and artificial neural network,” *Neural Comput. Appl.*, Nov. 2020, doi: 10.1007/s00521-020-05520-3.

[6] F.-K. Wang and T. Mamo, “Gradient boosted regression model for the degradation analysis of prismatic cells,” *Comput. Ind. Eng.*, vol. 144, p. 106494, Jun. 2020, doi: 10.1016/j.cie.2020.106494.

[7] Z. Xue, Y. Zhang, C. Cheng, and G. Ma, “Remaining useful life prediction of lithium-ion batteries with adaptive unscented kalman filter and optimized support vector regression,”

Neurocomputing, vol. 376, pp. 95–102, Feb. 2020, doi: 10.1016/j.neucom.2019.09.074.

[8] R. Wang and N. Chen, "Defect pattern recognition on wafers using convolutional neural networks," *Qual. Reliab. Eng. Int.*, vol. 36, no. 4, pp. 1245–1257, Jun. 2020, doi: 10.1002/qre.2627.

[9] S. Xiang, Y. Qin, C. Zhu, Y. Wang, and H. Chen, "Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction," *Eng. Appl. Artif. Intell.*, vol. 91, p. 103587, May 2020, doi: 10.1016/j.engappai.2020.103587.

[10] L. Ren, X. Cheng, X. Wang, J. Cui, and L. Zhang, "Multi-scale Dense Gate Recurrent Unit Networks for bearing remaining useful life prediction," *Futur. Gener. Comput. Syst.*, vol. 94, pp. 601–609, May 2019, doi: 10.1016/j.future.2018.12.009.

[11] A. Al-Dulaimi, S. Zabihi, A. Asif, and A. Mohammadi, "A multimodal and hybrid deep neural network model for Remaining Useful Life estimation," *Comput. Ind.*, vol. 108, pp. 186–196, Jun. 2019, doi: 10.1016/j.compind.2019.02.004.

[12] A. Khorram, M. Khalooei, and M. Rezghi, "End-to-end CNN + LSTM deep learning approach for bearing fault diagnosis," *Appl. Intell.*, vol. 51, no. 2, pp. 736–751, Feb. 2021, doi: 10.1007/s10489-020-01859-1.

[13] Y. Chen, G. Peng, Z. Zhu, and S. Li, "A novel deep learning method based on attention mechanism for bearing remaining useful life prediction," *Appl. Soft Comput.*, vol. 86, p. 105919, Jan. 2020, doi: 10.1016/j.asoc.2019.105919.

[14] B. Wang, Y. Lei, N. Li, and T. Yan, "Deep separable convolutional network for remaining useful life prediction of machinery," *Mech. Syst. Signal Process.*, vol. 134, p. 106330, Dec. 2019, doi: 10.1016/j.ymssp.2019.106330.

[15] X. Li, X. Jia, Y. Wang, S. Yang, H. Zhao, and J. Lee, "Industrial Remaining Useful Life Prediction by Partial Observation Using Deep Learning With Supervised Attention," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 5, pp. 2241–2251, Oct. 2020, doi: 10.1109/TMECH.2020.2992331.

[16] H. Li, W. Zhao, Y. Zhang, and E. Zio, "Remaining useful life prediction using multi-scale deep convolutional neural network," *Appl. Soft Comput.*, vol. 89, p. 106113, Apr. 2020, doi: 10.1016/j.asoc.2020.106113.

[17] M. Xia, X. Zheng, M. Imran, and M. Shoaib, "Data-driven prognosis method using hybrid deep recurrent neural network," *Appl. Soft Comput.*, vol. 93, p. 106351, Aug. 2020, doi: 10.1016/j.asoc.2020.106351.

[18] Z. Yang, J. Zhang, Z. Zhao, Z. Zhai, and X. Chen, "Interpreting network knowledge with attention mechanism for bearing fault diagnosis," *Appl. Soft Comput.*, vol. 97, p. 106829, Dec. 2020, doi: 10.1016/j.asoc.2020.106829.

[19] T. Xia, Y. Song, Y. Zheng, E. Pan, and L. Xi, "An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation," *Comput. Ind.*, vol. 115, p. 103182, Feb. 2020, doi: 10.1016/j.compind.2019.103182.

[20] R. Shang, J. He, J. Wang, K. Xu, L. Jiao, and R. Stolkin, "Dense connection and depthwise separable convolution based CNN for polarimetric SAR image classification," *Knowledge-Based Syst.*, vol. 194, p. 105542, Apr. 2020, doi: 10.1016/j.knosys.2020.105542.

[21] G. Huang, Y. Zhang, and J. Ou, "Transfer remaining useful life estimation of bearing using depth-wise separable convolution recurrent network," *Measurement*, vol. 176, p. 109090, May 2021, doi: 10.1016/j.measurement.2021.109090.

[22] A. Elsheikh, S. Yacout, and M.-S. Ouali, "Bidirectional handshaking LSTM for remaining useful life prediction," *Neurocomputing*, vol. 323, pp. 148–156, Jan. 2019, doi: 10.1016/j.neucom.2018.09.076.

[23] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *2008 International Conference on Prognostics and Health Management*, Oct. 2008, pp. 1–9, doi: 10.1109/PHM.2008.4711414.

[24] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2306–2318, Oct. 2017, doi: 10.1109/TNNLS.2016.2582798.

[25] G. Sateesh Babu, P. Zhao, and X.-L. Li, “Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, pp. 214–228.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 4, NO. 2, AUGUST 2021

Research Article

Diagnosing Hematological Disorders Using Deep Learning Method

Tuba Karagül Yıldız[1], Nilüfer Yurtay[2], Birgül Öneç[3]

[1]Corresponding Author; Sakarya University, Institute of Natural Sciences, Computer and Information Sciences, Turkey; tkaragul@sakarya.edu.tr; +90 506 518 76 28

[2]Sakarya University, Department of Computer Engineering, Turkey; nyurtay@sakarya.edu.tr;

[3]Duzce University, Medical School, Department of Internal Medicine, Turkey; birgulonec@gmail.com;

Received 5 December 2020; Revised 31 May 2021; Accepted 19 July 2021; Published online 31 August 2021

Abstract

Deciding on the diagnosis of the disease is an important step for treating the patients. Also, the numerical value of blood tests, the personal information of patients, and most importantly, an expert opinion is necessary to diagnose a disease. With the development of technology, patient-related data are obtained both rapidly and in large sizes. Deep learning methods, which can produce meaningful results by processing the data in raw form, are beginning to give results that are close to human opinion nowadays. The present work is aimed to develop a system that will enable the diagnosis of anemia in general practice conditions due to the increasing number of patients and the intention of the hospitals, as well as the difficulties in reaching the expert medical consultant. The main contribution of this work is to make a diagnosis like a doctor with the data as the way the doctor uses it. The data set was obtained from the actual hospital environment and no intervention, such as increasing or decreasing the number of data, increasing or decreasing the number of attributes, reduction, integration, imputation, transformation, or discretization, has been made on the incoming patient data. The original hospital data are classified for the diagnosis of anemia types and the accuracy of 84,97% achieved by using a deep learning algorithm.

Keywords: deep learning, anemia, classification

1. Introduction

The most common blood disease in Turkey and the world is the anemia group [1]-[4]. According to the World Health Organization, anemia is a condition in which the numbers of red blood cells and, consequently, oxygen-carrying capacities are inadequate to meet the body's physiological needs [2]. Anemia is also defined as a decrease in the concentration of erythrocyte mass or blood hemoglobin and hematocrit. Normal hemoglobin and hematocrit values vary according to age and sex. If hemoglobin and hematocrit values are below the lower limit of normal values for that age and sex, then there is anemia. In the study conducted by Kiassebaum et al. using data and resources from the World Health Organization's study on the global burden of disease 2010; 189 countries, 2 genders, and 20 different age groups were examined, and the global anemia frequency was calculated as 32.9%. Most commonly, anemia is seen in children under the age of five and women. Iron deficiency anemia is the most common type of anemia [3]. Anemia affects the quality of life significantly. It is not only a disease but also a symptom that accompanies many diseases like cancer and kidney or liver failure. It can be very vital to treat it in many cases. To treat a patient, first, it is necessary to decide on the disease.

Various methods have been developed to assist experts in making decisions. When diagnosing anemia types, information about the patient such as age, gender, symptoms, chronic diseases, and the results of the complete blood count analysis are also important factors together. In many methods recommended for the diagnosis of anemia, only a few of the results of complete blood count analysis are used instead of using whole results. The reason for this is, it is not desirable to increase the number of attributes in many classification methods. Attributes are defining each instance of the data set in different aspects. Especially in medical problems, there can be a lot of attributes and lots of missing values. In the deep learning method, it is not a problem being the high size of attributes or instances. Whereas data analysis with deep learning is giving more successful results than the classical methods with both raw and large-sized data.

To Cite This Article: T. Karagul Yildiz, N. Yurtay, B. Onec, “Diagnosing Hematological Disorders Using Deep Learning Method,” Sakarya University Journal of Computer and Information Sciences, vol. 4, no. 2, pp. 227–243, 2021. doi: 10.35377/saucis.04.02.836375

The method called deep learning is the name given to the deep neural network formed by adding many hidden layers to multilayer neural networks. Although it is not a new method, its use is becoming widespread and trendy nowadays. Deep learning algorithms have multi-input and multi-layered artificial neural network architecture. Deep neural networks' weight values are updated continuously for ensuring good results.

In this study, a multi-class probing solution was searched using deep learning architecture. It is aimed to develop a system that will enable the recognition of anemia in general practice conditions due to the increasing number of patients and the intention of the hospitals, as well as the difficulties in reaching the expert medical consultant. Running this system jointly with the tests required for diagnosing anemia in the primary health care services will help non-specialist personnel working in these health centers. At least based on this system, patients who need to be referred to the center can be selected faster and more accurately.

An overview of the problem of diagnosing anemia has been given in here, Introduction. The rest of the paper is organized as follows: Previous studies have been reviewed in Related Works. The anemia data used in this study has been described in Material and Method. Also, answers are given to questions like "what is anemia?" and "how to diagnose anemia?". Subsequently, methods used in the study are summarized and the architecture of the proposed deep learning system to diagnose anemia has been outlined. In Results, results of the models developed for this work are given and the evaluation has been made. Also, results are discussed. The motivation of this work and possible future works regarding this study has been given in Conclusion.

2. Related Works

Computer-aided decision-making and analysis is a widespread field in the medical area. A review of previous studies on the classification of anemia types was made, along with those conducted using similar methods but different data. One of the early studies about computer-aided anemia diagnosis is called PlanAlyzer. Beck et al., designed a computer-aided system for educational research in medical education. PlanAlyzer published in 1988 for diagnosis of heart disease [4] and anemia in 1989 [5] aimed to elucidate and criticize students' approach to diagnosing a widespread medical disorder. In a study published in 1993, Lyon et al. reported that this program was used for 7 years to teach the diagnosis of anemia and chest pain in cardiology and hematology departments of the Dartmouth School of Medicine after tests and assessments [6]. In 1960, Lipkin compared the data characteristics of hematological diseases and hospital data with the help of a digital computer. 49 patients and 20 diseases were selected, and hospital data were linked to the computer program. Differential diagnosis of hospital cases printed in written form [7]. In 1976, Engle and his colleagues introduced a computer program called HEME, which provided diagnostic analysis of hematological diseases to expert medical consultants. There are 40 hematological diseases in the system. The designed system was a rule-based system, and the Bayesian method was used [8]. Various algorithms have been developed to assist the doctor in the diagnosis of iron deficiency anemia, and their performances are successful [9] – [14]. Yavuz et al. conducted a study for the diagnosis of iron deficiency anemia in women. The data set used in the study includes six attributes including RBC, HGB, HCT, MCV, MCH, MCHC, and two classes and does not contain any missing values. Classification using ANNs and artificial immune system (AIS) was compared with that of KNN and the regression tree Gini algorithm. The performance of classification with the Gini-based decision tree method trained by the artificial immune system was more successful than the classification performance of the KNN method and ANNs [14]. Sanap et al. devised a classification that measured the severity of anemia using complete blood count reports and the C4.5 decision tree and support vector machine algorithms with the WEKA data mining tool. They included the ten numerical attributes of age, white blood cell count (WBC), hemoglobin (HGB), red blood cell count (RBC), hematocrit (HCT), mean cellular volume (MCV), mean cellular hemoglobin (MCH), mean cellular hemoglobin concentration (MCHC), red cell distribution width (RDW) and platelet count (PLT) and four classes of anemia types: normocytic (anemia of chronic disease), microcytic (iron deficiency and thalassemia), macrocytic (Vitamin B12 and folate deficiency) and microcytic (renal anemia). The success rate of the C4.5 decision tree algorithm was 99.42%, which surpassed the support vector machines with a success rate of 88.13% [15]. In the study conducted by Amin and Habib, the full blood count parameters of WBC, RBC, HGB, HCT, MCV, MCHC, PLT, neutrophils (NEUT), lymphocytes (LYMP), monocytes (MONO), eosinophils (EO), and basophils (BO) and interpretation values of age were used as the data input. The classes included chronic anemia, eosinophilia, microcytic hypochromic anemia, normocytic anemia, neutrophil leukocytosis, neutrophil, unknown findings, and high erythrocyte sedimentation rate (ESR). They used the J48 decision tree, multi-layered perceptron, and naive Bayes as classifiers and achieved success rates of 97.16%, 86.55%, and 70.28%, respectively [16]. Iron deficiency anemia and thalassemia are two types of microcytic anemia that are at risk of confusion [17]. In a research article, a differential diagnosis of microcytic anemia was made with discriminant analysis using a training set consisting of 200 beta-thalassemia cases, 65 alpha-thalassemia cases, 170 iron deficiency anemia cases, and 45 cases having both iron deficiency anemia and beta-thalassemia [18]. Jamei and Talarposhti developed an artificial neural network (ANN) model with pattern-based input selection for iron deficiency anemia and β-thalassemia carriage discrimination. This method consisted of the decision-making ability of the ANNs combined with that of a human expert. Using complete blood count results, they devised a coefficient rule base and determined the multilayer perceptron neural network’s input according to the calculated similarity. When compared with the performances of various authors using ANFIS, ANN, MLP, SVM, RBF, PNN, and KNN, their method was shown to have achieved the highest accuracy rate of 99.5% [19]. In 2015, Kishore et al. published a study by determining the age, gender, HGB, MCV, MCH, HCT values as input, and iron deficiency and Vitamin B12 deficiency as output. They have developed a threaded ID3 approach by examining ID3 and non-threaded ID3 decision tree algorithms as methods. With the 480 data, they tested the system with both threaded and non-threaded ID3 and Gini algorithms and reported that the method they found was usable [20]. Shaik and Subashini presented a fuzzy logic approach for anemia diagnosis. They have use HGB, HCT, MCV, MCHC, WBC, Reticulocyte, Total Iron Binding Capacity (TIBC), Serum iron, and HSWC (hyper segmented white cells) laboratory test results as input parameters. As output, they use six anemia types which are Aplastic, Sideroblastic, Megaloblastic, Chronic, Myelophthisic, and Iron deficiency anemias [21]. Dalvi and Vernekar have made a study to determine the most suitable method to classify Red Blood Cells for anemia diagnosis. They use five ensemble learning methods which are AdaBoost, bagging, stacking, voting, and Bayesian boosting, and four classifiers which are the k-nearest neighbor, naïve Bayes, decision tree, and artificial neural network [22]. Belginova et al present a rule-based approach to the diagnosis of iron deficiency anemia. They have made a decision support system for expert medical consultants which includes patient data like identification, social status, medical history, complaints or sensations, medical indicators, and statistical information on the disease. By using these data an expert can decide the disease more accurately [23]. Dimauro et al. have made a study to predict the Hemoglobin value of patients with a non-invasive device that can analyze the image of the conjunctival region. They have tested the KNN classifier on 113 persons and obtained good results [24]. Using the complete blood count parameters Hasani and Hanani classified four different diseases which are Iron deficiency anemia, alpha-thalassemia trait, beta-thalassemia trait, and healthy. In this study, the combination of J48 and Naïve Bayes methods was found to be the highest success [25]. Guncar et al used machine learning methods in the diagnosis of hematological diseases. The data they use consists of laboratory blood tests of patients in the hematology clinic. SVM, Naive Bayes, and Random Forest were used as machine learning algorithms. They used ROC analysis for performance evaluation. They declared that the random forest algorithm gave the most effective result [26]. Ayyıldız and Tuncer used machine learning techniques consisting of SVM and KNN for the differential diagnosis of iron deficiency anemia and beta-thalassemia. They used 7 attributes which are RBC indices as input. They used the NCA algorithm for feature selection. They stated that they are 97% successful using ROC analysis as performance evaluation [27].

Deep learning methods also provide successful results in this respect. In a study published by Schmidhuber in 2015, deep learning methods, which have been popular in pattern recognition and machine learning in recent years, are examined and the history of shallow and deep architectures is explained [28]. Liu et al. explored deep learning architectures such as Auto Encoder, Convolutional Neural Network (CNN), Deep Belief Network, and Restricted Boltzmann Machines, and explained how they were used in speech recognition, pattern recognition, and computer vision areas, in their work [29]. Deep learning methods are often used in the analysis of medical images. In a study by Litjens et al performed in 2017, studies in medical fields such as brain, eye, chest, skeleton, and pathology that use images in important tasks such as image classification, object recognition, and segmentation are surveyed [30]. Miotto et al proposed a method of non-consultant deep feature learning, modeling clinical prediction using electronic health records. They called their work the deep patient, and it contains raw data such as clinical notes, diagnosis, medications, and laboratory tests. Using this data, all the diseases that each patient has had over time are labeled. Thus, by reviewing the patients' past records, both diseases have been labeled and possible future diseases can be estimated [31]. Yu et al stated that they obtained higher accuracy in a shorter time by using a deep learning algorithm for white blood cell classification [32]. Xu et al made a study to diagnose sickle cell disease. They have developed a system to examine RBC (Red Blood Cell) images and used CNN (Convolutional Neural Networks) to classify RBC [33]. Kimura et al developed a deep-learning-based image analysis system to make a differential diagnosis of aplastic anemia and MDS. They divided the data set into 17 cell types and 97 abnormal morphological features. Then they created a CNN module to classify them. They used the ROC curve for performance measurement and the AUC (area under the curve) was found as 0.99 [34]. Alsheref and Gomaa conducted a study on blood disease prediction with machine learning methods. They developed a new data set consisting of 668 patients' blood tests, containing 28 parameters and 4 classes. They used the WEKA tool for applying machine learning algorithms and found the highest result of 98.16% with the logit boost method [35]. Çil, Ayyıldız, and Tuncer have developed a decision support system to make a differential diagnosis of iron deficiency anemia and beta-thalassemia. They performed a 95.59% successful classification using logistic regression, KNN, SVM, ELM, and RELM algorithms. They used HGB, RBC, HCT, MCV, MCH, MCHC, and RDW parameters of 342 patients [36]. Varghese conducted a study highlighting the importance of machine learning methods in the classification of blood cells and blood diseases. Varghese analyzed lots of articles and claims that CNN and SVM methods give high success results [37]. Alzubaidi et al conducted a study to diagnose sickle cell anemia by classifying RBC shapes. They recognized RBCs as 3 classes as normal, sickle cell, and other blood content. In their study, they proposed 3 different deep learning models. They used transfer learning and data augmentation since there was no training data. They stated that their models made high accuracy classification [38]. Devunooru et al. conducted a study for use in the diagnosis of brain tumors. They performed the segmentation process of my brain MRI images using deep learning methods and obtained successful results [39]. Göceri has conducted a study that will automate the classification of dermatological diseases. Within the scope of the study, he finds and identifies the lesioned areas on the skin from color digital photographs and classifies them [40]. Kılıçarslan et al. developed a deep learning-based hybrid method for the classification of nutritional anemia diseases. At the end of the classification process, the data were divided into 3 classes: HGB-anemia, nutritional anemia, and patients without anemia. They enriched the stacked autoencoder and CNN methods, deep learning algorithms, genetic algorithms, and successful results [41]. Yeruva et al., in their study, determined the type of deterioration of hemoglobin structure from the shapes of red blood cells using deep neural network methods. They classified it according to the shape of the cells as sickle cell, thalassemia, and normal. As a method, they compared the multilayer perceptron method with other machine learning methods and stated that they were successful [42]. Gupta et al. conducted a study that detected the presence of covid 19 pneumonia, a new disease that affects millions of people around the world. They determined whether the patient had covid 19 from chest x-ray images with the deep convolution network method and said that they were successful [43]. Venugopalan et al. conducted a study to determine the stage of Alzheimer's disease. They emphasized the success of deep learning methods by comparing deep learning methods with other machine learning methods in identifying various regions of the brain on brain MRI images [44]. Atila et al. conducted a study that used deep learning architecture to classify diseases observed in the leaves of plants [45].

When these studies are examined, it is seen that studies are carried out to diagnose one or more general types of anemia like microcytic, normocytic, and macrocytic anemias in [15], [16] or thalassemia and iron deficiency anemia in [17], [18] and [19]. Although various research were carried out up to date, it has not been encountered that diagnosing 14 different anemia types related to study in the literature. Thus, in this study, a multi-class probing solution for diagnosing 14 different types of anemia which are described in WHO’s International Disease Classification (ICD) Codes were investigated by using deep learning architecture. Besides, in most of the works above, the attributes used are mostly limited to a few blood parameters only. In this study, 30 different attributes that are used by the expert medical consultant to diagnose the diseases have been used. Also, the data used in this study are completely original and include age, gender, chronic diseases, and symptoms as well as blood parameters.

3. Material and Method

The present study, it is aimed to develop a system that will enable the recognition of anemia in general practice conditions due to the increasing number of patients and the intention of the hospitals, as well as the difficulties in reaching the expert medical consultant. For this purpose, real patient data and the advice of an expert medical consultant are needed. For the diagnosis of anemia, information like which feature is needed, which method should be followed to diagnose which disease has been determined with the expert medical consultant, and the necessary data required could only be obtained in this way from the ethics committee. Once ethical approval is obtained, data were transferred from a clinical database to the program interface shown in Figure 1.

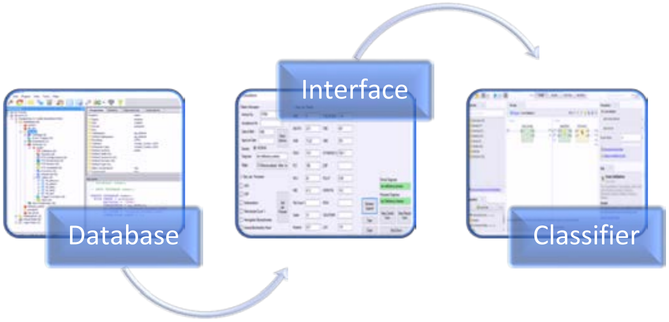


Figure 1. The Design of the Proposed System.

As seen in Figure 1 real patient data was transferred from a database to the program interface. With the help of the interface, data are given to the deep learning environment to classify anemia types. The transactions made during this process do not involve any interference with the data. Only data stored in multiple tables in the database are combined into a single table. After the data were obtained and prepared with the help of the program interface, a deep learning model was developed for the classification process. To demonstrate, the diagram of the present study is shown in Figure 2.

In this step, the training and the testing process were conducted for the deep learning method. Finally, the developed model was recorded and tested on new data and the performance was evaluated by accuracy, classification error, kappa, weighted mean recall and precision, absolute and relative errors, root means squared error (RMSE) and Receiver Operating Characteristic (ROC) analysis method. The flowchart diagram of the entire study is shown in Figure 2.

3.1 Dataset Definition

The data used in the present study are actual patient data obtained from the Duzce University Research and Practice Hospital with permission from the ethics committee. To obtain the approval of the ethics committee, every detail had to be determined. Therefore, all the attributes of the data were to be announced to the committee. So, it has not been needed for any pre-processing other than digitizing data and no intervention was required. Since there was no process for changing the content and quality of the data, in the present study a completely original dataset was used in which not even any numerical intervention was made. The data obtained by the approval of the ethics committee includes 34 attributes. Three of which are located in the raw dataset and which enable us to organize our data are not used as attributes. These are archive number, accept number, and approval date. Archive number is a unique attribute to define the patient in the hospital records. By the accept number and the approval date of the patient, it can be reached that how often the same patient applies to the clinic. These data are not included in the dataset. All the attributes used in the dataset are shown in detail in Table 1.

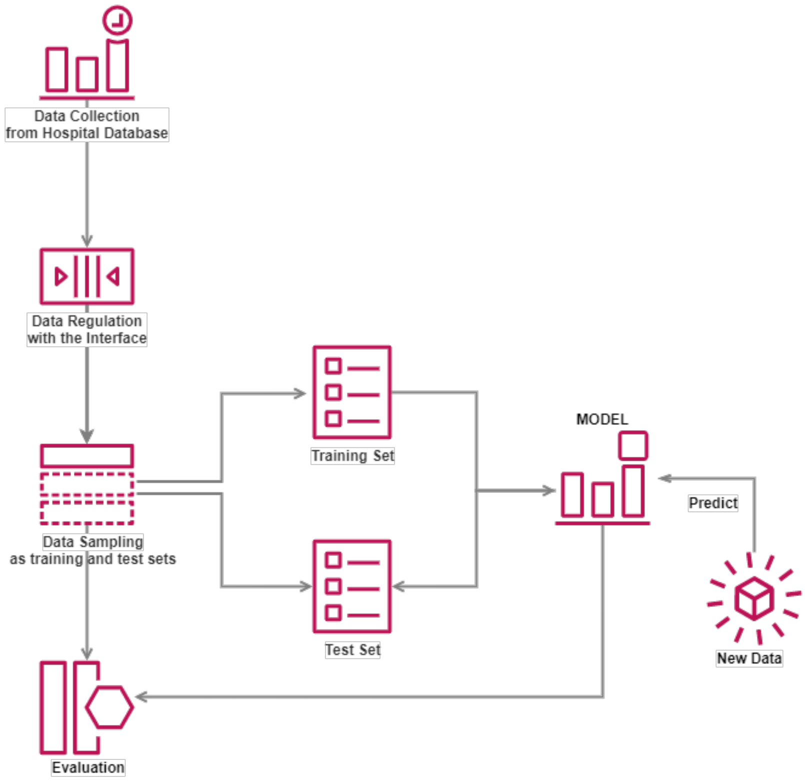


Figure 2. The Flow Chart of the Whole Study.

Information such as age, gender, presence of symptoms, and chronic diseases of the patients are attributes that play an important role in determining the type of anemia. Other Information contains the medical history of the patient. Bilirubin values in blood analysis are used to assess liver and gall bladder functions. C Reactive Protein (CRP) provides information about the presence of inflammation in the body. Iron value in the blood is used in the evaluation of all kinds of anemia, iron deficiency, and iron poisoning. Ferritin value is used in the diagnosis of iron deficiency anemia, chronic disease anemia, and thalassemia. It is also important for following iron loading treatment. Folate means folic acid value in the blood and is used in the evaluation of megaloblastic and macroscopic anemia. Also, it is used to observe the treatment of folate deficiency anemia. Hematocrit (HCT) value shows the amount of hemoglobin and erythrocytes existence in the blood. Hemoglobin (HGB) value shows the total amount of hemoglobin existence in blood. It is the first value to investigate in complete blood count parameters that indicate anemia. Creatinine value in the blood is used in the evaluation of kidney functions. Lactate dehydrogenase (LDH) is an enzyme that helps the body produce energy and is released into the bloodstream in situations such as sickness and injury in that cells are damaged. Thus, the value of the LDH in the blood increases. It is an important parameter for the detection of hemolytic anemia from anemia types. Mean Cell Hemoglobin (MCH) shows the total amount of hemoglobin in erythrocytes. Mean Cell Hemoglobin Concentration (MCHC) is the percentage of hemoglobin concentration in erythrocytes. Mean Corpuscular Volume (MCV) is the average size of red blood cells carrying oxygen. NEUT# value is the number of neutrophils in the blood. NEUT% value is the percentage of neutrophils in the blood. Procalcitonin (PCT) is a hemogram parameter that indicates the percentage of thrombocytes in the blood. Platelet Dispersion Width (PDW) is another hemogram parameter that shows the distribution range of the sizes of thrombocytes. The lower value of PDW means that the sizes of the platelets are very similar, and the higher value of PDW means that the sizes of the platelets are very different in size from each other. PLT value is the number of platelets that allow blood to clot. Red Blood Cell (RBC) is the amount of erythrocytes presence in red blood cells. Red cell Distribution Width (RDW) shows the distribution width of erythrocytes in the blood. Total iron-binding capacity (TIBC) and unbound iron-binding capacity (UIBC) are also important parameters to diagnose anemia types. Vitamin B12 is an essential vitamin for hematopoiesis and normal neuronal functions. In case of a lack of Vitamin B12, Vitamin B12 deficiency anemia may be considered. The White blood cell (WBC) value is the number of leucocytes in the white blood cells which is responsible for the body's defense and immunity system [46]-[50].

Table 1 List of the Attributes in the Dataset

Attribute Name	**Type**	**Min.**	**Max.**	**Avg.**
Age	Numeric	4	91	54,54
Gender	Polynomial	-	-	-
Chronic Disease	Polynomial	-	-	-
Symptoms	Polynomial	-	-	-
Other Information	Polynomial	-	-	-
CRP (C Reactive Protein)	Numeric	0	27,60	1,24
D. Bilirubin (Direct Bilirubin)	Numeric	0	2,45	0,22
Iron	Numeric	4	377	79
Ferritin	Numeric	0	899,40	111,76
Folate	Numeric	2,61	25,20	10,64
HCT (Hematocrit)	Numeric	11	64,50	35,48
HGB (Hemoglobin)	Numeric	3,80	22,90	11,79
I. Bilirubin (Indirect bilirubin)	Numeric	-0,13	5,01	0,49
Creatinine	Numeric	0,23	8	0,92
LDH (Lactate Dehydrogenase)	Numeric	57	3685	227,63
MCH (Mean Cell Hemoglobin)	Numeric	13,99	45,20	27,54
MCHC (Mean Cell Hemoglobin Concentration)	Numeric	25,60	38,20	33,13
MCV (Mean corpuscular volume)	Numeric	49	126,60	82,87
NEUT# (Neutrophils)	Numeric	0	47,83	4,12
NEUT%	Numeric	0	95,20	57,77
PCT	Numeric	0	1015	1,10
PDW	Numeric	8,60	31,50	14,82
PLT (Platelets)	Numeric	2	1239	259,12
RBC (Red Blood Cell)	Numeric	1,24	8,84	4,33
RDW (Red cell Distribution Width)	Numeric	11,20	38,20	17,21
T. Bilirubin (Total Bilirubin)	Numeric	-0,02	5,73	0,72
TIBC (Total iron-binding capacity)	Numeric	104	697	354,14
UIBC (Unbound iron-binding capacity)	Numeric	-9	676	275,14
Vitamin B-12	Numeric	30	1941	495,33
WBC (White blood cell)	Numeric	0,70	431,33	8,07
Diagnosis	Polynomial	-	-	-

There are 1477 data in the data set. The distribution of these data according to diagnosis is given in Table 2. As it is seen, in Duzce which is a province in Turkey's western Black Sea region, the distribution

Table 2 Distribution of the Diagnoses in the Dataset

ICD 10 Code	**Diagnosis**	**Count**	**Percentage**
D50	Iron deficiency anemia	468	31,69 %
D64	Anemia	283	19,16 %
D63	Anemia of chronic disease	241	16,32 %
---	Other	132	8,94 %
D46	Myelodysplastic Syndrome	127	8,60 %
D56	Thalassemia	68	4,60 %
D51	Vitamin B12 deficiency anemia	38	2,57 %
D59	Hemolytic anemia	35	2,37 %
D50-D51	Iron and vitamin B12 deficiency anemia	29	1,96 %
D50-D51-D52	Iron, folate and vitamin B12 deficiency anemia	16	1,08 %
D50-D52	Iron and folate deficiency anemia	13	0,88 %
D51-D52	Folate and vitamin B12 deficiency anemia	12	0,81 %
D52	Folate deficiency anemia	11	0,74 %
D57	Carriage of thalassemia	4	0,27 %
	Total	1477	100 %

of diseases associated with anemia is not regular and balanced. There are 977 female and 500 male patients in the data set. Women are known to have a high prevalence of anemia. These data also confirm the nature of this situation. Iron deficiency anemia, which constitutes 31,69 % of the dataset, is the most common type of anemia seen in the Duzce region. The least common disease is the thalassemia trait. However, it should be kept in mind that to obtain these data for use in our study, ICD codes are limited between D50 and D64.9, and attributes are limited to 34 different features. Therefore, the attributes needed to diagnose anemia-associated diseases were selected by the expert medical consultant's recommendation.

The anemia types listed here are the 14 most common types of anemia in the region of Duzce in Turkey where data are taken. As the data are taken from the hospital, all the patients in the dataset have at least one hematologic disorder. The "other" patient group does not consist of healthy individuals. For this reason, it is not considered a control group. Since they are suffering from other hematologic disorders outside the anemia group, the use of the "other" term was considered appropriate. In the last case, classification with 31 attributes including the diagnosis column shown in Table 2 was performed.

3.2 Deep Learning Method

Artificial neural networks are one of the most widely used classification methods in the literature. It is an artificial learning model designed by inspiring the nervous system in living things. The inspired nervous system consists of neurons and synaptic connections of neurons. When the mathematical model of a neuron is designed, the synaptic connections are represented by connection weights. Each neuron should have input connections, bias values, activity level, output connections, and an output value. Equation (1) shows the output connection of a neuron where $\boldsymbol{a_i}$ indicates the input value, $\boldsymbol{w_i}$ indicates the weight value of each input, and b indicates the bias value for the neuron.

$$\boldsymbol{f(x) = f(\sum_{i=1} w_i a_i + b} \tag{1}$$

The output is created by making calculations in the equation for each input value to a neuron. The neuron value is reflected in the transfer function f (x) as the sum of the input signals multiplied by the weights. This output is passed through the activation function and transmitted to the next layer. The learning capacity of an artificial neuron is determined by arranging the weights of the chosen learning algorithm [51,52].

The artificial neural network usually consists of an input layer, a hidden layer, and an exit layer. In deep networks, both the number of these layers and the number of neurons in the layers should be higher. The deep network used in this study is based on a multi-layer feed-forward neural network trained by stochastic gradient descent using a backpropagation algorithm. The proposed model includes an input layer containing 30 neurons, 5 hidden layers each containing 50 neurons, and an output layer containing 14 neurons. The network structure of the proposed model is shown in Figure 3.

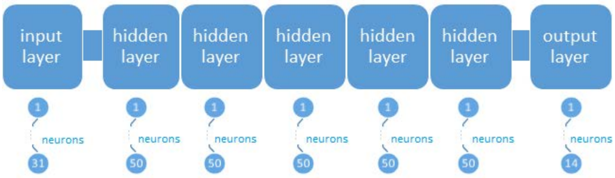


Figure 3. The Basic Structure of the Proposed Model.

The number of hidden layers can be very numerous. The activation functions and numbers of the neurons in the hidden layer are determined by the structure of the problem [53]. For this work, hyperbolic tangent function (tanh), rectifier linear unit, maxout, and exponential rectifier linear unit functions are selected as the activation functions. The hyperbolic tangent function is a version of scaled and shifted sigmoid function. Equation (2) shows the sigmoid function where x indicates inputs.

$$\boldsymbol{\delta(x) = \frac{1}{1 - e^{-x}}} \qquad (2)$$

The rectifier linear unit function selects the maximum values of (0, x) where x is the input value. The maxout function selects the maximum coordinate of the input vector. The exponential rectifier linear unit function is the exponential version of the rectifier linear unit function.

The network structure of the proposed method is seen in Figure 4. In the present study, 5 hidden layers are used, each of the hidden layers consists of 50 neurons inside.

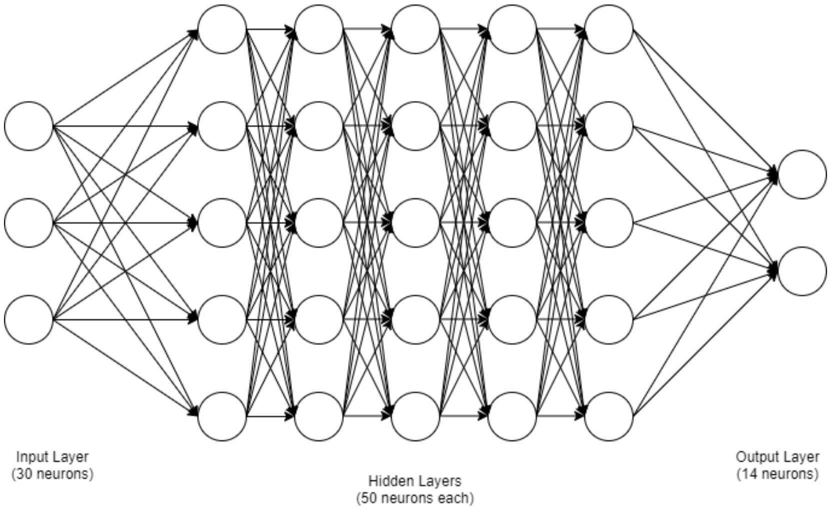


Figure 4. The Network Structure of the Proposed Method.

Deep learning is a popular method in recent years because it gives successful results on big and raw data without pre-processing. With the development of technology, there is a lot of data both in terms of size and quality and it takes time to perform pre-processing or cleaning on these data. Classical methods often cause memory errors due to the size of the data and are not sufficient. In such situations, deep learning methods are preferred, which process data instantly and produce results, and thus use less memory. While deep learning is widely used in the areas of object recognition and image processing which are using big sizes of memory it is becoming increasingly common to use with numerical or polynomial data in the important scientific fields that affect human health, such as medicine. This study contributes to the advancement of the use of deep learning in medical areas.

3.3 Evaluation

For the present study, k-fold cross-validation method was used to evaluate the proposed method. In the k-fold cross-validation method, the dataset was divided into k different subsets. When a group is a test set, the remaining groups are used as training sets. In this way, all the combinations were tested so that each of the k data subsets is used as once a test set. As a result, a performance value was found by taking the average of each result.

Receiver operating characteristic (ROC) analysis is used for performance measurement in this study. The ROC analysis is an effective method for measuring the performance of machine learning and data mining techniques [54]. ROC curve is used to evaluate the equilibrium between accuracy, sensitivity and specificity. Accuracy, sensitivity and specificity values are basically calculated as the formulas given in Equations (3), (4) and (5) respectively.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \qquad (3)$$

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{4}$$

$$Specificity = \frac{TN}{(TN + FP)} \tag{5}$$

Expressions which are used in the formulas (3), (4), and (5) are given below:

TP (True Positive): Number of the instances where the predicted value is positive, and the real value is positive.

TN (True Negative): Number of the instances where the predicted value is negative, and the real value is negative.

FP (False Positive): Number of the instances where the predicted value is positive, and the real value is negative.

FN (False Negative): Number of the instances where the predicted value is negative, and the actual value is positive.

The ROC curve of the presented method used in the present study is shown in the Figure 5.

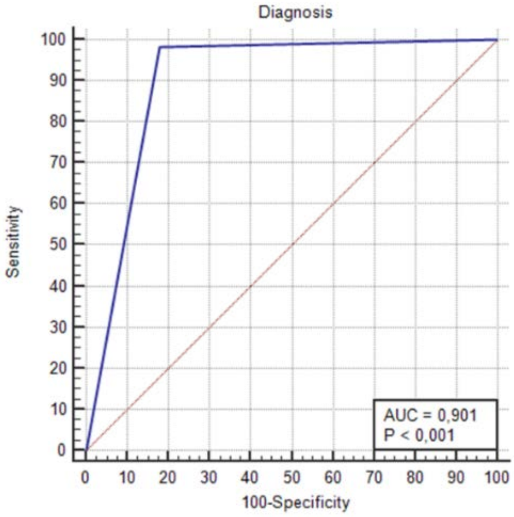


Figure 5. The ROC Curve of the Presented Method.

The area under the ROC curve is an important parameter to measure the success of classification methods. The closer this value, which defines as AUC -Area under the curve, to 1, the more accurate the prediction has been made. For the success of the classification to be acceptable, it is desired to form a curve in the upper left part of the red line in the graph shown in Figure 5. Otherwise, the classification is considered unsuccessful. If the AUC value is 1, all classes are correctly estimated. If the AUC value is 0, it means that no class is correctly predicted [54].

The model also evaluated by accuracy, classification error, kappa, weighted mean recall and precision, absolute and relative errors, root means squared error methods. Accuracy is the percentage of correctly predicted data. Classification error is te percentage of the data that predicted incorrectly. Kappa statistics is calculated as the Equation (6):

$$\kappa = \frac{P(a) - P(r)}{1 - P(r)} \tag{6}$$

Here, ***P(a)*** represents correctly predicted data, while ***P(r)*** represents correctly predicted data by chance. If κ is 1, it means that the data was predicted completely correctly, while κ is 0 means that the data was predicted correctly by pure chance. The precision is the rate at which the predicted positive class is actually positive. The recall is the ratio of correctly predicted true positives [55]. The absolute and relative errors and root mean squared error (RMSE) are also significant methods used in performance measurements. In the present study, performance evaluation is made with the methods mentioned above and the results are given in the next section with Tables 6-8.

4. Results and Discussion

In the presented study, a system that can diagnose anemia like an expert medical consultant is designed. For this purpose, the deep learning method is tested with the original dataset including real patient data. The data used in the present study are original patient data obtained from the Research and Practice Hospital of Duzce University with permission taken from the ethics committee of the hospital. The attributes used are shown in Table 1. The information about how often the same patient applies to the clinic is reached by using the date of approval, the archive, and the acceptance numbers of patients, which are not included as attributes. Also, information such as age, gender, symptoms, and other chronic diseases of the patients are attributes that play an important role in determining the type of anemia. Other attributes in Table 1 are obtained from laboratory test results of complete blood count. Diagnosis decision is made by the expert medical consultant. To determine the presence of anemia, firstly the hemoglobin (HGB) value is examined. If HGB is less than 13 g / dL for men and 12 g / dL for women, then there is anemia. In the next step, the Mean Corpuscular Volume (MCV) value is examined. Anemia can be macrocytic if MCV is greater than 100, normocytic if MCV is between 80 and 100, or microcytic if MCV is less than 80. After identifying the first phase the expert may require further investigation and advanced tests for a definite diagnosis. For example, if the iron value is low and creatinine value is normal, iron deficiency anemia can be considered as the diagnosis. If creatinine value is also low, then anemia of chronic disease can be considered as the diagnosis. Detailed types or causes of anemia are shown in Figure 6.

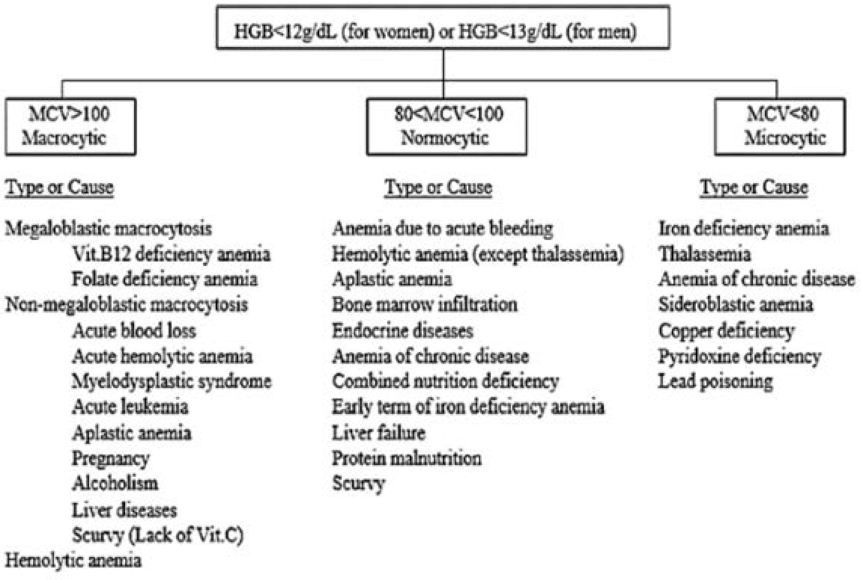


Figure 6. The Classification of Anemia According to the Erythrocyte Morphology [50].

The present study aimed to diagnose 14 different types of hematological disorders, most of which are anemia groups. The diseases and ICD-10 codes are seen in Table 2. The total numbers of attributes are 31, the total numbers of instances in the data set are 1477 and the total classes are 14. When the data is classified by using the deep learning algorithm, the results shown in Table 3 are obtained.

As seen in Table 3, the highest accuracy and minimum classification error achieved by using 10-fold cross-validation with the deep learning algorithm. In the cross-fold validation method, the data set is divided into k different subsets. When a subset is a test set, the remaining k-1 subsets are set as a training set. In this way, all the combinations are tested so that each of the k subsets is once a test set, and as a result, a performance value is found by taking the average of each result. Also, the ROC analysis method was used for performance measurement. The ROC curve of the proposed method is seen in Figure 5. Table 4 and Table 5 show the confusion matrixes of the proposed method.

Table 3. Performance of the proposed method.

Deep Learning	**Accuracy**	**Error**
Split validation (0,7)	81,00 %	19,00 %
Cross-validation (5-fold)	83,07 %	16,93 %
Cross-validation (7-fold)	83,28 %	16,72 %
Cross-validation (10-fold)	*84,97 %*	*15,03 %*
Cross-validation (12-fold)	84,29 %	15,71 %
Cross-validation (15-fold)	84,50 %	15,50 %

Table 4. Confusion matrix of 14 classes (Split Validation).

	True														
Pred.	Ane.	IDA	CoT	F-B12-DA	ACD	B12-DA	Other	I-F-DA	MDS	I-B12-DA	HA	I-F-B12-DA	FDA	Thal.	class prec.
Ane.	60	13	0	1	7	3	1	1	0	0	2	0	1	1	66.6
IDA	14	121	1	0	3	0	1	1	0	2	0	1	1	3	81.7
CoT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0
F-B12-DA	0	0	0	3	0	0	0	0	0	0	0	0	0	0	100
ACD	6	2	0	0	58	0	1	0	0	0	1	0	0	1	84.0
B12-DA	1	2	0	0	0	7	0	0	0	0	2	0	0	0	58.3
Other	3	0	0	0	4	1	37	0	0	0	0	0	0	0	82.2
I-F-DA	0	0	0	0	0	0	0	2	0	0	0	0	0	0	100
MDS	0	0	0	0	0	0	0	0	38	0	0	0	0	0	100
I-B12-DA	0	0	0	0	0	0	0	0	0	7	0	0	0	0	100
HA	0	0	0	0	0	0	0	0	0	0	5	0	0	0	100
I-F-B12-DA	0	0	0	0	0	0	0	0	0	0	0	4	0	0	100
FDA	1	0	0	0	0	0	0	0	0	0	0	0	1	0	50.0
Thal.	0	2	0	0	0	0	0	0	0	0	0	0	0	15	88.2
class recall	70.5	86.4	0.0	75.0	80.5	63.6	92.5	50.0	100	77.7	50.0	80.0	33.3	75.0	

Abbreviations: IDA - Iron deficiency anemia, Ane. - Anemia, ACD - Anemia of chronic disease, Other - Other, MDS - Myelodysplastic Syndrome, Thal. - Thalassemia, B12DA - Vitamin B12 deficiency anemia, HA - Hemolytic anemia, I-B12-DA - Iron and vitamin B12 deficiency anemia, I-F-B12-DA - Iron, folate, and vitamin B12 deficiency anemia, I-F-DA - Iron and folate deficiency anemia, F-B12-DA - Folate and vitamin B12 deficiency anemia, FDA - Folate deficiency anemia, CoT - Carriage of thalassemia.

As seen in Table 4, MDS is the best-predicted class with a 100% accuracy rate, and CoT is found as the worst predicted class with a 0% accuracy rate. The average accuracy rate is 81% for split validation. And as seen in Table 5, again MDS is the best-predicted class with a 98,43% accuracy rate and FDA is found as the worst predicted class with a 0% accuracy rate. The average accuracy rate for 10-fold cross-validation is 84,97%.

Table 5. Confusion matrix of 14 classes (10-fold cross-validation).

	True														
Pred.	Ane.	IDA	CoT	F-B12-DA	ACD	B12-DA	Other	I-F-DA	MDS	I-B12-DA	HA	I-F-B12-DA	FDA	Thal.	class prec.
Ane.	215	28	0	2	20	5	12	2	1	3	2	1	3	3	72.3
IDA	42	430	1	1	12	8	3	2	1	9	1	1	2	3	83.3
CoT	0	0	3	0	0	0	0	0	0	0	0	0	0	0	100
F-B12-DA	0	0	0	7	0	0	0	0	0	0	0	0	0	0	100
ACD	17	4	0	1	206	1	4	0	0	0	1	0	0	0	88.0
B12-DA	1	2	0	0	0	22	0	0	0	1	0	0	0	0	84.6
Other	6	2	0	1	0	2	112	0	0	0	1	0	1	0	89.6
I-F-DA	0	0	0	0	0	0	0	9	0	0	0	0	0	0	100
MDS	0	2	0	0	2	0	1	0	125	0	0	0	1	0	95.4
I-B12-DA	0	0	0	0	0	0	0	0	0	16	0	0	0	0	100
HA	0	0	0	0	0	0	0	0	0	0	30	0	0	0	100
I-F-B12-DA	0	0	0	0	0	0	0	0	0	0	0	14	0	0	100
FDA	1	0	0	0	0	0	0	0	0	0	0	0	4	0	80.0
Thal.	1	0	0	0	1	0	0	0	0	0	0	0	0	62	96.8
class recall	75.9	91.8	75.0	58.3	85.4	57.8	84.8	69.2	98.4	55.1	85.7	87.5	36.3	91.1	

Table 6 shows the success rates obtained by using the Tanh, Rectifier, Maxout, and ExpRectifier activation functions after 50 epochs.

Table 6. Success rates of the method after 50 epochs.

50 epochs	Tanh	Rectifier	Maxout	ExpRectifier
Accuracy (%)	81,38	78,61	75,70	81,04
Classification error (%)	18,62	21,39	24,30	18,96
Kappa	0,768	0,734	0,697	0,765
Recall (%)	66,48	58,74	54,82	65,89
Precision (%)	71,90	62,96	60,45	70,69
Absolute error	0,195	0,221	0,254	0,195
Relative error (%)	19,54	22,07	25,38	19,46
RMSE	0,413	0,446	0,479	0,413

Table 7 shows the success rates obtained by using the Tanh, Rectifier, Maxout, and ExpRectifier activation functions after 100 epochs.

Table 7. Success rates of the method after 100 epochs.

100 epochs	Tanh	Rectifier	Maxout	ExpRectifier
Accuracy (%)	84,63	78,33	75,84	81,18
Classification error (%)	15,37	21,67	24,16	18,82
Kappa	0,809	0,730	0,699	0,767
Recall (%)	70,87	60,77	57,51	66,47
Precision (%)	76,62	66,54	62,71	71,72
Absolute error	0,165	0,227	0,249	0,200
Relative error (%)	16,50	22,72	24,86	19,96
RMSE	0,380	0,444	0,469	0,422

Table 8 shows the success rates obtained by using the Tanh, Rectifier, Maxout, and ExpRectifier activation functions after 150 epochs.

Table 8. Success rates of the method after 150 epochs.

150 epochs	Tanh	Rectifier	Maxout	ExpRectifier
Accuracy (%)	82,80	80,10	75,09	81,51
Classification error (%)	17,20	19,90	24,91	18,49
Kappa	0,787	0,753	0,689	0,769
Recall (%)	71,25	65,13	54,30	66,07
Precision (%)	76,48	69,70	60,23	73,40
Absolute error	0,175	0,206	0,254	0,194
Relative error (%)	17,49	20,60	25,43	19,36
RMSE	0,395	0,425	0,478	0,414

As seen in the tables above, the highest success was obtained with the tanh activation function. In addition, another parameter that affects success is the epoch number. Since it is understood from the tables that increasing the number of epochs does not increase the success, on the contrary, it decreases the success. From all reasons above, it was understood that the most suitable parameters for the data used in this study were 100 epochs with the tanh activation function.

When the studies in the literature are analyzed, it is seen that there is no study diagnosed for 14 different types of anemia using deep learning methods. Yavuz et al [14] conducted a study on iron deficiency anemia in women. In their study, they decided whether there was iron deficiency using 6 attributes. Although it has a high success rate, only a single disease was diagnosed. In our study, it is a strong aspect that classes are more, and the method used is up to date. Yu et al [32] classified white blood cells using deep learning methods in their study. Xu et al [33], on the other hand, classified the red blood cells using deep learning methods. High successes have also been achieved in the mentioned studies. Although the data used are different from our study in terms of being images. Besides that, their studies are similar to our study because they both diagnose blood diseases, and the methods they use are deep learning methods. Hasani and Hanani [25] used complete blood count parameters similar to our study and they have obtained successful results. In this study, iron deficiency and alpha and beta-thalassemia traits were diagnosed. In our study, 14 different anemia diagnoses were made. Besides, the data we use include data such as the patient's age, gender, and other chronic diseases, as well as complete blood count parameters. These data were included in the study because they are the data affecting the doctor's decision for diagnosis. In this respect, our study stands in a different place from other studies.

In the present study, a deep learning-based system was developed to diagnose 14 different types of anemia. The data set was obtained from the actual hospital environment and no intervention, such as increasing or decreasing the number of data, increasing, or decreasing the number of attributes, reduction, integration, imputation, transformation, or discretization, has been made on the incoming patient data. Therefore, since there is no process to change the content and quality of the data, an original dataset is used in this work which is not made any intervention numerically. In addition, the use of real patient data makes this study important. When the dataset is obtained, the data are classified with the deep learning method at the next stage. The deep learning method which is quite popular in the literature has been implemented and significant results have been found. The 10-fold cross-validation method was used during classification. In this method, the data set is divided into 10 different subsets. When a group is a test set, the remaining 9 groups are set as a training set. In this way, all the combinations are tested so that each of the 10 subsets is once a test set, and as a result, an impressive performance value is found by taking the average of each result.

ROC analysis method was used for performance measurement besides accuracy, classification error, kappa, weighted mean recall and precision, absolute and relative errors, root means squared error methods. An impressive success rate was obtained, as seen in the performance evaluation. It is reasonable that this success will increase when data cleaning and pre-processing are made. In future work, we aim to increase success by using different algorithms of deep learning methods. Also, it is aimed to make various hybrid methods by making additions to these algorithms or using different machine learning methods together, thus, to increase the success ratio. Moreover, by expanding the dataset, hematological diseases other than anemia could also be included in these studies.

5. Conclusion

Anemia is a very common disease affecting the quality of life and with the appropriate treatment, the standard of living of the patient will improve. It is obvious that the first step in the treatment is the correct diagnosis. In the present study, the 14 different types of anemia most encountered in Duzce Province were classified by the deep learning method. One of the very important points of this study is, this is the first time with this data and this method is used together. No comparison was made with previous studies because the data used for this study is completely original and with this data, no method has been tried before. Therefore, we believe that any comparison with other studies will not be ethical. For those who want to work in different ways, data can be shared with another permission of the ethics committee which has not been appealed yet.

The strength of the present study was 14 different anemia types are diagnosed like a doctor by the deep learning algorithm. The limitation of this study is that the entire dataset consists of data from individuals with hematologic diseases. In addition, anemia is also a symptom that can accompany many other diseases at the same time. Despite this limitation, it is obvious that the classifier performances obtained are acceptable. This study contributes positively to the literature. Our results showed that it is possible to diagnose like a doctor with the raw data by using the presented method.

Furthermore, in order to improve success, future works should focus on combining different methods and on developing new hybrid methods. Moreover, by expanding the dataset, hematological diseases other than anemia could also be included in our studies.

Acknowledgments

Funding- This work was supported by the Research Fund of Sakarya University, Turkey, under Project Number: 2015-50-02-010. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Ethical Approval- The data used in this study were taken from the Duzce University Research and Practice Hospital, Turkey, with the approval of the Duzce University Ethics Committee (Decision number 2014/82).

References

[1] T.Vos, D. Flaxman et al., Years lived with disability (YLDs) for 1160 sequelae of 289 diseases and injuries 1990–2010: A systematic analysis for the Global Burden of Disease Study 2010, The Lancet, 380, 2163-2196, DOI: 10.1016/S0140-6736;12;61729-2, 2012.

[2] Hemoglobin concentrations for the diagnosis of anemia and assessment of severity Vitamin and Mineral Nutrition Information System. Geneva, World Health Organization, Available: http://www.who.int/vmnis/indicators/haemoglobin.pdf (2011).

[3] N.J. Kiassebaum, R. Jasrasaria et al., A systematic analysis of global anemia burden from 1990 to 2010. Blood, 123, 615-624, DOI: 10.1182/blood-2013-06-508325, 2014.

[4] J.R. Beck, J.R. Bell, F.Hirai, J.J. Simmons, H.C. Jr. Lyon, Computer-Based Exercises in Cardiac Diagnosis (PlanAlyzer), Proc. Annu. Symp. Computer Applications in Medical Care, Nov 9, 403-408, PMCID: PMC2245328, 1988.

[5] J.R. Beck, J.F. O'Donnell, F. Hirai, J.J. Simmons, J.C. Healy, H. C. Jr. Lyon, Computer-based exercises in anemia diagnosis (PlanAlyzer), Methods Inf. Med., 28:4, 364-369, PMID:2695787, 1989.

[6] H. C. Jr. Lyon, J.R. Bell, J.F. O'Donnell, F. Hirai, J.C. Healy, J.R. Beck, The PlanAlyzer Cases for Teaching Clinical Reasoning: A Demonstration of the Cases, Discussion of the Research & Development Process, Lessons Learned and Strategies for Introducing Computer-Based Programs into Medical School Courses as a Vehicle for Curriculum Reform, Proc. Annu. Symp. Computer Applications in Medical Care, PMCID: PMC2850769, 1993.

[7] M. Lipkin, Correlation of Data with a Digital Computer in the Differential Diagnosis of Hematological Diseases, IRE Transactions on Medical electronics, 243-246, 1960.

[8] R. L. Engle, B. J. Flehinger, S. Allen, R. Friedman, M. Lipkin, B. J. Davis, L. L. Leveridge, HEME: A Computer Aid to Diagnosis of Hematologic Disease, Bulletin of the New York Academy of Medicine, 52:5, 584–600, 1976.

[9] I. Azarkhish, M.R. Raoufy, S. Gharibzadeh, Artificial Intelligence Models for Predicting Iron Deficiency Anemia and Iron Serum Level Based on Accessible Laboratory Data, J Med Syst, 36, 2057-2061, DOI: 10.1007/s10916-011-9668-3, 2012.

[10] Z. Yılmaz, M.R. Bozkurt, Determination of Women Iron Deficiency Anemia Using Neural Networks, J Med Syst, 36, 2941-2945, DOI: 10.1007/s10916-011-9772-4, 2012.

[11] A. Yılmaz, M. Dağlı, N. Allahverdi, A Fuzzy Expert System Design for Iron Deficiency Anemia, IEEE 7th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, DOI: 10.1109/ICAICT.2013.6722707, 2013.

[12] W. Reinisch, M. Staun, S Bhandari., M. Munoz, State of the iron: How to diagnose and efficiently treat iron deficiency anemia in inflammatory bowel disease, Journal of Crohn's and Colitis, 7, 429-440, 2013.

[13] Ş. Doğan, I. Türkoğlu, Iron-Deficiency Anemia Detection from Hematology Parameters by Using Decision Trees, International Journal of Science & Technology, 3:1, 85-92, 2008.

[14] B.C. Yavuz, T. Karagül Yıldız, N. Yurtay, Z.Yılmaz, Comparison of K Nearest Neighbors and Regression Tree Classifiers Used with Clonal Selection Algorithm to Diagnose Hematological Diseases, AJIT-e: Online Academic Journal of Information Technology, 5-16, DOI: 10.5824/1309-1581.2014.3.001. x, 2014.

[15] S. A. Sanap, M. Nagori, V. Kshirsagar, Classification of Anemia Using Data Mining Techniques, Presented in SEMCCO 2011, Berlin Heidelberg, Part II, 2011.

[16] N. Amin, A. Habib, Comparison of Different Classification Techniques Using WEKA for Hematological Data, Am J Eng Res, 4:3, 55-61, e-ISSN: 2320- 0847, 2015.

[17] I. Yılmaz, Demir eksikliği anemisi ile beta talasemi minörün ayırıcı tanısında eritrosit indekslerin rolü, Expertise Thesis, Dept. of Internal Medicine, Faculty of Medicine, Pamukkale University, Denizli, Turkey, 2010.

[18] E. Urrechaga, U. Aguirre, S. Izquierdo, Differential Diagnosis of Microcytic Anemia, Anemia, DOI: 10.1155/2013/457834, 2013.

[19] M. K. Jamei, K. M. Talarposhti, Discrimination between Iron Deficiency Anaemia (IDA) and β - Thalassemia Trait (β-TT) Based on Pattern-Based Input Selection Artificial Neural Network (PBIS- ANN), J Adv Comp Res, 7: 4, 55-66, pISSN: 2345-606x, eISSN: 2345-6078, 2016.

[20] R. Kishore, K.P. Rao, G.R.S. Murthy, Performance Evaluation of Entropy and Gini using Threaded and Non-Threaded ID3 on Anaemia Dataset, Presented at Fifth International Conference on Communication Systems and Network Technologies, IEEE, 2015.

[21] M. F. Shaik, M. Subashini, Anemia Diagnosis by Fuzzy Logic Using LabVIEW, Presented at IEEE International Conference on Intelligent Computing and Control (I2C2), DOI: 10.1109/I2C2.2017.8321790, 2017.

[22] P. T. Dalvi, N. Vernekar, Anemia Detection using Ensemble Learning Techniques and Statistical Models, Presented at IEEE International Conference on Recent Trends in Electronics Information Communication Technology, May 20-21, 2016.

[23] S. Belginova, I. Uvaliyeva, A. Ismukhamedova, Decision Support System for Diagnosing Anemia, Presented at 4th International Conference on Computer and Technology Applications, DOI: 10.1109/CATA.2018.8398684, 2018.

[24] G. Dimauro, D. Caivano, F. Girardi, A New Method and a Non-Invasive Device to Estimate Anemia Based on Digital Images of the Conjunctiva, IEEE Access-Special Section on Human-Centered Smart Systems and Technologies, DOI: 10.1109/ACCESS.2018.2867110, 2018.

[25] M. Hasani, A. Hanani, Automated Diagnosis of Iron Deficiency Anemia and Thalassemia by Data Mining Techniques, International Journal of Computer Science and Network Security, 17:4, 326-331, 2017.

[26] G. Gunčar, M. Kukar, M. Notar, M. Brvar, P. Černelč, M. Notar, M. Notar, An application of machine learning to haematological diagnosis, Scientific Reports, 8:411, 1-12, DOI:10.1038/s41598-017-18564-8, 2018.

[27] H. Ayyıldız, S. A. Tuncer, Determination of the effect of red blood cell parameters in the discrimination of iron deficiency anemia and beta thalassemia via Neighborhood Component Analysis Feature Selection-Based machine learning, Chemometrics and Intelligent Laboratory Systems, 196, 1-8, DOI: https://doi.org/10.1016/j.chemolab.2019.103886, 2020.

[28] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Networks, 61, 85–117. DOI: 10.1016/j.neunet.2014.09.003, 2015.

[29] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F. E. Alsaadi, A survey of deep neural network architecture and their applications, Neurocomputing, 234, 11–26, DOI: 10.1016/j.neucom.2016.12.038, 2017.

[30] G. Litjens et al., A survey on deep learning in medical image analysis, Medical Image Analysis, 42, 60–88, DOI: 10.1016/j.media.2017.07.005, 2017.

[31] R. Miotto, L. Li, B. A. Kidd, J. T. Dudley, Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records, Scientific Reports, 6:26094,1-10. DOI: 10.1038/srep26094, 2016.

[32] W. Yu, J. Chang, C. Yang, L. Zhang, H. Shen, Y. Xia, J. Sha, Automatic Classification of Leukocytes Using Deep Neural Network, IEEE 12th International Conference on ASIC, DOI: 10.1109/ASICON.2017.8252657, 2017.

[33] M. Xu, D.P. Papageorgiou, S.Z. Abidi, M. Dao, H. Zhao, G.E. Karniadakis, A deep convolutional neural network for classification of red blood cells in sickle cell anemia, PLoS Computational Biology, 13:10, DOI: https://doi.org/10.1371/journal.pcbi.1005746, 2017.

[34] K. Kimura, Y. Tabe, T. Ai, I. Takehara, H. Fukuda, H. Takahashi, T. Naito, N. Komatsu, K. Uchihashi, A. Ohsaka, A novel automated image analysis system using deep convolutional neural networks can assist to differentiate MDS and AA, Scientific Reports, 9, 1-9, DOI: https://doi.org/10.1038/s41598-019-49942-z, 2019.

[35] F. K. Alsheref, W. H. Gomaa, Blood Diseases Detection using Classical Machine Learning Algorithms, International Journal of Advanced Computer Science and Applications, 10:7, 77-81, 2019.

[36] B. Çil, H. Ayyıldız, T. Tuncer, Discrimination of β-thalassemia and iron deficiency anemia

through extreme learning machine and regularized extreme learning machine-based decision support system, Medical Hypotheses, 138, 1-6, DOI: https://doi.org/10.1016/j.mehy.2020.109611, 2020.

[37] N. Varghese, Machine Learning Techniques for the Classification of Blood Cells and Prediction of Diseases, International Journal of Computer Science Engineering, 9:1, 66-75, 2020.

[38] L. Alzubaidi, M. A. Fadhel, O. Al-Shamma, J. Zhang, Y. Duan, Deep Learning Models for Classification of Red Blood Cells in Microscopy Images to Aid in Sickle Cell Anemia Diagnosis, Electronics, 9:427, 1-20, DOI:10.3390/electronics9030427, 2020.

[39] S. Devunooru, A. Alsadoon, P. W. C. Chandana, A. Beg, Deep learning neural networks for medical image segmentation of brain tumours for diagnosis: a recent review and taxonomy, Journal of Ambient Intelligence and Humanized Computing (2021) 12:455–483, https://doi.org/10.1007/s12652-020-01998-w

[40] E. Goceri, Deep learning based classification of facial dermatological disorders, Computers in Biology and Medicine, Volume 128, January 2021, 104118, https://doi.org/10.1016/j.compbiomed.2020.104118

[41] S. KILICARSLAN, M. CELIK, Ş. SAHIN, Hybrid models based on genetic algorithm and deep learning algorithms for nutritional Anemia disease classification, Biomedical Signal Processing and Control, Volume 63, January 2021, 102231, https://doi.org/10.1016/j.bspc.2020.102231

[42] S. Yeruva, M. S. Varalakshmi, B. P. Gowtham, Y. H. Chandana, PESN. K. Prasad, Identification of Sickle Cell Anemia Using Deep Neural Networks, Emerging Science Journal (ISSN: 2610-9182) Vol. 5, No. 2, April, 2021, http://dx.doi.org/10.28991/esj-2021-01270

[43] A. Gupta, Anjum, S. Gupta, R. Katarya, InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using Chest X-ray, Applied Soft Computing, Volume 99, February 2021, 106859, https://doi.org/10.1016/j.asoc.2020.106859

[44] J. Venugopalan, L. Tong, H. R. Hassanzadeh, M. D. Wang, Multimodal deep learning models for early detection ofAlzheimer's disease stage, Scientific Reports, (2021) vol. 11, article number. 3254, https://doi.org/10.1038/s41598-020-74399-w

[45] Ü. Atila, M. Uçar, K. Akyol, E. Uçar, Plant leaf disease classification using EfficientNet deep learning model, Ecological Informatics, Volume 61, March 2021, 101182, https://doi.org/10.1016/j.ecoinf.2020.101182

[46] Haematology in Clinical Practice, Part 1: Erythrocyte Disorders, Chapter 2. Clinical Approach to Anemia, 5th ed., Güneş Medical Bookstore, ISBN: 978-975-277-404-9, 2012.

[47] B. Onec, Personal Lecture Notes of Birgül Öneç, Duzce University, Medical School, Department of Internal Medicine Sciences, 2017.

[48] R. S. Hillman, K. A. Ault, M. Leporrier, H. M. Rinder, Haematology in Clinical Practice. Part 1: Red Blood Cell Disorders, Chapter 2. Clinical Approach to Anemia, 5th ed., Istanbul, Turkey: Güneş Medical Bookstore, ISBN: 978-975-277-404-9, 2012.

[49] Guide to diagnosis and treatment of Erythrocyte Diseases and Hemoglobin disorders, Turkish Hematology Association, Version1-July2011, Available: www.thd.org.tr, 2011.

[50] R. Hoffman, E.J. Benz, L.E. Silberstein, H.E. Heslop, Weitz J.I., Anastasi J., Hematology: Basic Principles and Practice, 6th edition, ISBN: 978-1-4377-2928-3, Elsevier, 2013.

[51] A. E. Hassanien, E. T. Al-Shammari, N. Ghali, Computational Intelligence Techniques in Bioinformatics, Computational Biology and Chemistry, DOI: 10.1016/j.compbiolchem. 2013.04.007, 2013.

[52] E. Öztemel, Artificial Neural Networks, Papatya Press, Istanbul, Turkey, 2006. ISBN: 975-67-97-39-8.

[53] Rapidminer Tutuorial, 2017. [Online]. Available: https://docs.rapidminer.com/studio/operators/modeling/predictive/neural_nets/deep_learning.html [Accessed: 30.10.2017].

[54] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters, 27, 861-874, 2006.

[55] F.Ramzan, M.U.G.Khan, A.Rehmat, S.Iqbal, T.Saba, A.Rehman, Z.Mehmood, A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks, Journal of Medical Systems vol.44(37), 2020.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 4, NO. 2, AUGUST 2021

Research Article

Base Station Power Optimization for Green Networks Using Reinforcement Learning

Semih Aktaş[1], Hande Alemdar[2]

[1] Department of Computer Engineering, Middle East Technical University, Turkey; aktas.semih@metu.edu.tr;
[2] Corresponding Author; Department of Computer Engineering, Middle East Technical University, Turkey;
alemdar@metu.edu.tr; +90 312 210 55 91

Received 4 May 2021; Revised 5 July 2021; Accepted 2 August 2021; Published online 30 August 2021

Abstract

The next generation mobile networks have to provide high data rates, extremely low latency, and support high connection density. To meet these requirements, the number of base stations will have to increase and this increase will lead to an energy consumption issue. Therefore "green" approaches to the network operation will gain importance. Reducing the energy consumption of base stations is essential for going green and also it helps service providers to reduce operational expenses. However, achieving energy savings without degrading the quality of service is a huge challenge. In order to address this issue, we propose a machine learning based intelligent solution that also incorporates a network simulator. We develop a reinforcement-based learning model by using deep deterministic policy gradient algorithm. Our model update frequently the policy of network switches in a way that, packet be forwarded to base stations with an optimized power level. The policies taken by the network controller are evaluated with a network simulator to ensure the energy consumption reduction and quality of service balance. The reinforcement learning model allows us to constantly learn and adapt to the changing situations in the dynamic network environment, hence having a more robust and realistic intelligent network management policy set. Our results demonstrate that energy efficiency can be enhanced by 32% and 67% in dense and sparse scenarios, respectively.

Keywords: Green networking, reinforcement learning, deep deterministic policy gradient, LTE

1. Introduction

Next generation mobile networks have to meet requirements such as high data rates, extremely low latency, and connection density. Due to the rapid growth of telecommunications technology, energy consumption is also growing at a very fast rate [1, 2]. Mobile service providers are among the top energy consumers [3]. The increase in the energy consumption of mobile networks negatively affects the environment and causes higher operational expenses for mobile service providers. Therefore, "green network" approaches become more popular to reduce energy consumption as well as the cost [4, 5, 6].

In a 5G network, the number of base stations (BS) will increase significantly to meet 5G requirements. Consequently, the energy consumption problem will be more prominent. Luckily, with the development of software networks (SN), it is possible to dynamically configure cells to reduce power consumption when the traffic load is low. This dynamic configuration technique is known as the sleeping strategy or ON-OFF switching. The sleeping strategy is considered as an approach for energy saving [7, 8]. Therefore, advanced sleeping strategies need to be implemented for future green networks in order to achieve better efficiency without harming the network performance. Machine learning (ML) can be a remedy in that issue.

Mobile network function virtualization (NFV) can be applied over the core and the radio access network (RAN) [9]. This means that we can virtualize these modules and provide on-demand network functions in both of them. Because around 70%-80% network is consumed in RAN, network operators expand their investigations in virtualizing RAN in the future networks. In this work, we propose consolidating RAN functions using NFV inside multi-access edge clouds (MECs) to reduce the amount of energy consumption in the access network. Our proposed network architecture is shown in Figure 1.

To Cite This Article: S. Aktas, H. Alemdar, "Base Station Power Optimization for Green Networks Using Reinforcement Learning," Sakarya University Journal of Computer and Information Sciences, vol. 4, no. 2, pp. 244–265, 2021. doi: 10.35377/saucis.04.02.932709

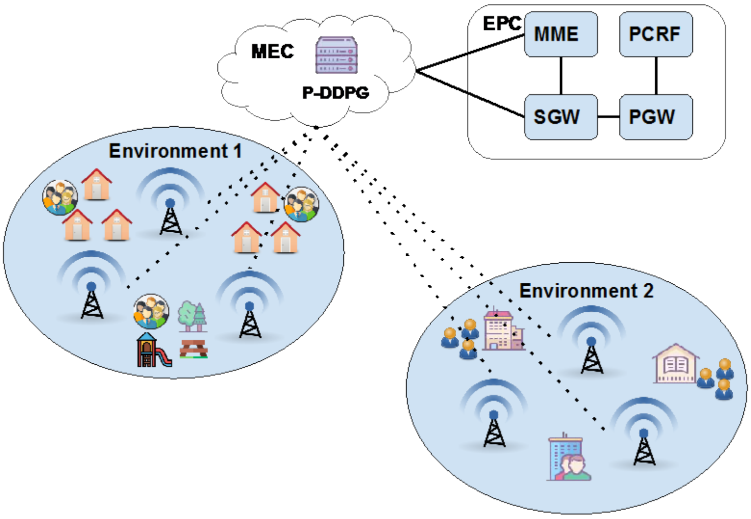


Figure 1 P-DDPG Network Architecture

ML provides a way of automatically learning about the environment by using historical data when it is challenging to construct and solve analytic models. Hence, ML is suitable for modelling stochastic environments such as the wireless networks [10]. With the development of SN, it is now possible the exercise the powerful capabilities of machine learning for network management in terms of intelligent decision making. That is why, recently, the usage of machine learning in the network management has become an active research area [11, 12, 13]. However, there are only a handful of studies that addresses the dynamic nature of the wireless networks.

In order to address the issue, we propose a reinforcement-based approach to employ an advanced adaptive sleeping strategy by gathering constant feedback from the network and continuing to learn under changing dynamic conditions such user requests, packet arrival time. We develop our solution basis on Deep Deterministic Policy Gradient (DDPG) algorithm [10] which is a type of reinforcement learning (RL) algorithm. We extend DDPG algorithm to work with multiple environments as parallel and we called it Parallel DDPG (P-DDPG). Our model reduces power consumption of a group of base stations while maintaining users' quality of service (QoS). In RL, the model continuously learns by taking some actions following a strategy and observing the outcomes of these actions to adjust its strategy over time. By taking many actions, the model learns to differentiate the good actions from the bad ones. This makes its policies evolve over time. In order to find novel potential good strategies, the model sometimes explores new horizons rather than sticking to the exploitation of what has been learnt all the time. This mechanism allows the learner to adapt to the changing conditions as well. This scheme is often considered similar to how a child learns by exploring her environment. Like a child, the RL model makes more mistakes at the beginning of the learning and when it becomes more mature the decisions made are more robust and correct. These initial phases of the learning can be problematic if we deploy the model in the real network environment. To address this issue, we use a system-level network simulator to create a dynamic network environment and we observe the outcomes our actions in this simulator. This allows us to learn a more robust model that captures realistic network dynamics rather than static assumptions about the network while preventing the real users suffering from bad decisions. After the model is mature enough, the learnt policy can be deployed in the real network controller safely. To the best of our knowledge, this is the first study that employs such a realistic scheme.

Our main contributions in this paper are:

- We developed P-DDPG algorithm, which enables DDPG to work for parallel environments (Section 2.2). This enabled us to run multiple environments to accelerate learning.
- We developed a machine learning model that reduces energy consumption while maintaining QoS parameters of users on a realistic simulation environment by using the P-DDPG algorithm (Section 2.3).

- Simulation parameters are given for future reproduce. Simulation results and detailed parameter analysis are presented (Section 3).

Our results show that it is possible to achieve up to 32% increase in the energy efficiency in a dense scenario and up to 67% increase in sparser scenarios while preserving user QoS parameters such as throughput and Signal to Interference Ratio (SINR).

2. Modelling P-DDPG for Energy Efficient Base Station Control

In this section, the evolution of DDPG algorithm, our extended DDPG model which is P-DDPG and the details of our RL scheme are described.

2.1 Background of Reinforcement Model

Reinforcement learning is a type of machine learning which is focused on goal-directed learning from interaction [14]. RL is an efficient method for sequential decision-making problems, making them ideal for network management [15]. In RL, the *learner agent* takes *actions*, and each action receives a *reward* as a feedback signal. The reward is positive if the outcome of the action is good in terms of the goal achievement and it is negative otherwise. Through this reward collection mechanism, the agent learns a *policy*, that is, the action sequences required to solve a problem. RL is widely used in dynamic environments where a *state* can be rewarded as positive or negative without analytically modelling the environment but making observations about the outcome instead.

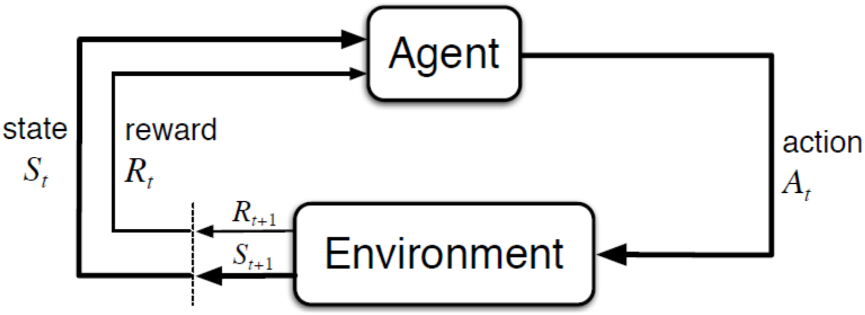


Figure 2 Overview of Reinforcement Learning Algorithms [12]

The general workflow of RL algorithms is summarized in Figure 2. The agent takes an action at time t, (A_t), according to the observation in the same time step, (S_t). The environment performs the action and returns the observation (S_{t+1}) and the feedback signal (R_{t+1}). The feedback signals are used to update the policy, i.e., action decision model.

RL has been applied in different ways over time. Q-learning is a one type of RL algorithm. It is based on Q-tables, where rows represent the states and columns represent the actions. All of the action decisions are made by looking at the Q-table, which contains the whole policy. Q-learning considers the possible future reward when rewarding the instant status [16]. It is formulated as follows:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right], \tag{1}$$

where α represents learning rate and γ represents the discount factor. Each cell in the Q-table is created by considering the maximum expected future reward. Q-learning suffers from the curse of dimensionality because of the need to create a table for each state-action pair. Although it is a convenient way to use when a problem has discrete state space or discrete action space, Q-table cannot be created for continuous state spaces or continuous action spaces.

The significant performance improvement in RL comes with deep reinforcement learning, which is also called as the “Deep Q Network” (DQN) [17]. Deep deterministic policy gradient (DDPG) algorithm is developed for continuous control with deep reinforcement learning [10]. DDPG is a combination of the deterministic policy gradient (DPG) algorithm [18] with the DQN. In this work, we employ the DDPG approach since our space is continuous. In the next section, we provide the details of our approach and our extension to the original DDPG approach.

2.2 P-DDPG Model

As we mentioned previously, the proposed algorithm is implemented in a MEC in order to virtualize RAN functions. In this section, we explain the P-DDPG algorithm in detail.

Traditionally, DDPG algorithm works with a single environment. As can be seen in Figure 2, the operating time is restricted by the Agent's response time or the runtime of the environment. In our case, environment's runtime is considerably slower than the Agent's response time. Therefore, we extend DDPG algorithm to work with multiple environments as parallel and we call it Parallel DDPG (P-DDPG).

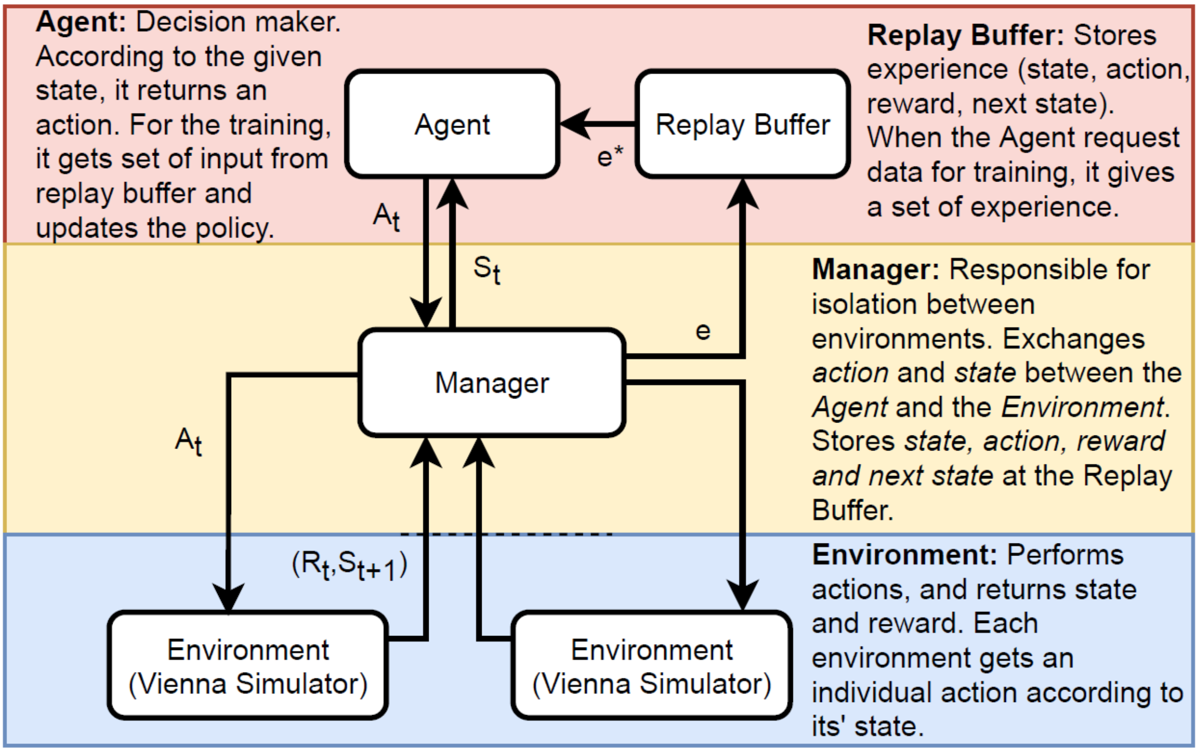

Agent: Decision maker. According to the given state, it returns an action. For the training, it gets set of input from replay buffer and updates the policy.
Agent
Replay Buffer
e*
Replay Buffer: Stores experience (state, action, reward, next state). When the Agent request data for training, it gives a set of experience.
At
St
e
Manager
Manager: Responsible for isolation between environments. Exchanges *action* and *state* between the *Agent* and the *Environment*. Stores *state, action, reward and next state* at the Replay Buffer.
At
(Rt,St+1)
Environment (Vienna Simulator)
Environment (Vienna Simulator)
Environment: Performs actions, and returns state and reward. Each environment gets an individual action according to its' state.

Figure 3 Structure of P-DDPG Algorithm with Parallel Environment

Figure 3 shows that the structure of P-DDPG algorithm which is implemented in MEC in order to virtualize RAN functions including BSs transmit power adaptation. Implementing P-DDPG in MEC can provide higher perspective over the network with respect to BSs, while actions and decisions will be made quicker in comparison to the core, due to the location of MEC in the network.

In P-DDPG, the manager is responsible for environment virtualization and memorizing operations. It virtualizes each environment and creates a private channel between the agent and each environment. Through the private channels, each environment works independently from each other. The manager is responsible for taking an environment state and sending them to the agent. Agent decides an action according to the state and sends an action back to the manager. The manager sends back the action to a related environment. Meanwhile, manager stores state (S_t), action (A_t), reward (R_t) and next state (action result, S_{t+1}) in replay buffer [17]. Each replay buffer entry is called as experience (e) and the agent uses replay buffer entities for batch training. It trains deep neural networks by using samples from replay buffer. The DDPG algorithm [10] psudocode is followed for policy updating and state-action exchange.

Environment paralleling is a novel approach for DDPG algorithms, and it reduces the convergence time of DDPG algorithm when environment operation time significantly greater than agent's response time.

Each environment configuration is called an episode. Episode length refers to a number of state-action exchange which also defined as CTI. In a single episode, when a certain number of consecutive rewards are negative, we stop the environment's episode and start a new episode. We called this method as a consecutive negative reward check (CNRC). CNRC method is used because when actions cannot improve environment states, and negative rewards continue, eventually negative state-action pairs dominate replay buffer. By using CNRC, the number of positive and negative state-action pairs are balanced in replay buffer. It increases the convergence time of the model.

The use of mathematically modelled environments based on strong assumptions in model training is a common but unrealistic method. It is not possible to use models that trained in hypothetical environments in real systems. Modelling with realistic environment is a challenging problem. We use Vienna Simulator [19], known as advanced realistic network simulator, as an environment. We make some changes to the simulator so that Vienna Simulator can work with P-DDPG. We develop connectors for information exchange between Vienna Simulator and P-DDPG. Moreover, we develop our custom indicators which are defined as overall statuses (OSs). With these modifications, Vienna Simulator has capability of working with P-DDPG. Thanks to Vienna Simulator, our model is trained in a realistic environment that accommodates many real-life factors such as noise, interference, shadowing, fading.

2.3 Reinforcement Learning Definitions

In our model, user states are measured at each transmission time interval (TTI) which is 1 millisecond in 4G mobile networks. The user behavior and requirements change during time. Sometimes the users are idle, sometimes they actively use the communication channels, therefore each TTI status does not represent the status of the network. Also, after changing base stations' transmission powers, we need to wait a while before observing action result. Therefore, the environment and agent state exchange is performed at certain time intervals, and we call this interval as the communication time interval (CTI). In other words, each RL cycle period that is depicted in Figure 2 is one CTI.

The user denoted by *u* are assigned to the base station that denoted by *bs*. *U* and *BS* refers to the set of users and set of base stations with respectively. User status for specific CTI is represented as different notations. For CTI at *k*, user wideband SINR is called as $SINR^k(u)$. The amount of data (MB) that transmitted by the user at specific CTI is represented as $\Psi^k(u)$. User active TTI count for CTI^k notated as $\varphi^k(u)$.

State: State is a representation of the instant status. Environment creates state vector by using status of network and BSs. Environment state at CTI^k is defined as:

$$S^k = \left(OS_i^k, S1_j^k, S2_j^k, S3_j^k, S4_j^k\right), \tag{2}$$

where $i = 1, 2, .., 4$ and $j = 1, 2, .., N$. *N* is the number of base stations. *OS* represents the overall status of network. Users' wideband SINR and transmission values are used while formulating overall statuses of network and base station. Overall wideband SINR equation is

$$OS_1^k = \frac{\sum_{u\in U} SINR^k(u) \times \phi^k(u)}{\sum_{u\in U} \phi^k(u)} \tag{3}$$

In Equation 3, users' wideband SINR average value is calculated when users are actively using communication channels. Overall network throughput is formulated as

$$OS_2^k = \frac{\sum_{u\in U} \Psi^k(u)}{t} \tag{4}$$

The total amount of data that served to users is divided by CTI length, *t*. Users' average throughput according to their active TTI count is formulated as

$$OS_3^k = \frac{\sum_{u\in U} \Psi^k(u)}{\sum_{u\in U} \Phi^k(u)} \tag{5}$$

Users who are close to base stations can mislead Equation 4 but averaging with active TTI count will normalize throughput considering idle users. Average throughput per user is formulated as

$$OS_4^k = \frac{\sum_{u\in U} \Psi^k(u)}{|U| \times t} \tag{6}$$

Overall statuses (*OS*$_i$, *i*=1,2,..,4) give general information about the network configuration which consists a group of base station and users. Overall SINR value in Equation 3 shows the possibility of transmitting data while users' throughput sum in Equation 4 shows the actual usage. Equation 5 considers idle users. Users' throughput sum in Equation 4 and users' throughput average in Equation 6 change when the number of users change. Therefore, each OS value has a unique usage and give information about network. OS values are used as QoS parameters.

Our model also considers base stations' statuses when deciding their power level. BSs' powers scaled to [0,1] according to

$$S1_j^k = \frac{P^k(BS_j)}{P_{\max}}, \tag{7}$$

where $P^k(b)$ refers to the base station *b*'s power at CTI^k and P_{max} refers to maximum power of macro BS. The number of users assigned to each base station at CTI^k is also used as a status of base station. The equation is

$$S2_j^k = |U_j^k|, \tag{8}$$

where U_j^k refers to the set of users which are assigned to base station *j* at CTI *k*. The status of BS with respect to the user average wideband SINR is

$$S3_j^k = \frac{\sum_{u\in BS_j} SINR^k(u) \times \phi^k(u)}{\sum_{u\in BS_j} \phi^k(u)}, \tag{9}$$

where $u \in BS_j$ refers to users that assigned to BS_j. Equation 9 gives information about users' wideband SINR average value according to their active TTI count. Users' average throughput according to their active TTI count is formulated as

$$S4_j^k = \frac{\sum_{u\in BS_j} \Psi^k(u)}{\sum_{u\in BS_j} \phi^k(u)}, \tag{10}$$

where $u \in BS_j$ refers to users that assigned to BS_j. The state vector is a combination of the status of network and BSs, therefore, an increase in the number of base stations causes the state vector to grow.

Action: The action at CTI^k is represented as $a^k = (\Delta P_1, \Delta P_2, \ldots, \Delta P_N)$ where ΔP_i refers to the power change for base station *i* and *N* refers to the number of base stations. Actions are scaled to [-1, 1]. The action dimension is related with the number of base stations. The formula for new transmit powers of base stations is that:

$$P^{k+1}(BS_i) = P^k(BS_i) + \left[a_i^k \times P_{max}\right] \tag{11}$$

where P_{max} refers to maximum power of macro BS.

Reward: Rewards are decided according to the long-term goal that the model should satisfy. Our long-term goal is maintaining QoS while reducing energy consumption. Hence, the proposed model decides rewards by considering the overall status of network and energy consumption. Each OS creates its own reward according to pre-defined threshold values as follows

$$R_{OS_i}^k = \begin{cases} 1, & if\ th_i^u < OS_i \\ -\frac{th_i^u - OS_i}{th_i^u - th_i^l}, & if\ th_i^u \geq OS_i > th_i^l \\ -1, & otherwise \end{cases} \tag{12}$$

where $th^u{}_i$ and $th^l{}_i$ refers to upper and lower threshold values for OS_i. When OS_i is below $th^u{}_i$ then negative reward appears. To calculate threshold values, the network is observed without taking any action and we call these observations as a baseline. The long-term goal is maintaining the overall status of the network same as the baseline. Therefore, we set threshold values according to the baseline's average QoS parameters. Upper and lower threshold values are used to scale negative rewards to [0, –1].

Energy consumption is also important when we consider a reward. When the proposed model satisfies QoS, then according to energy consumption, it gains a positive reward. The environment uses an average of scaled powers divided by the number of the base station and uses the gain as a positive reward

$$R_{EC}^{k} = 1 - \frac{\sum_{bs \in BS} S_1^k(bs)}{|BS|} + \epsilon, \tag{13}$$

where R_{EC} stands for energy consumption reward and ε is a small positive value. The ε value ensures that even if all BSs are configured as maximum power, R_{EC} will always positive. Reducing the current transmit power of the base stations increases R_{EC}. Environment sends only one reward to agent, and this reward is calculated as

$$R^k = min\left(R_{\mathrm{OS}_1}^k, R_{\mathrm{OS}_2}^k, R_{\mathrm{OS}_3}^k, R_{\mathrm{OS}_4}^k, R_{EC}^k\right) \tag{14}$$

At CTI^k, if any of OS_i is below $th^u{}_i$ then environment sends negative reward to agent. Otherwise, since ${R_{EC}}^k$ is always positive, environment sends positive reward to agent. Environment sends positive reward if and only if user QoS parameters, which are OS1 – 4, are higher than threshold values.

We use energy efficiency (EE) as a measure metric. The equation of energy efficiency is

$$EE = \frac{\sum_{u \in U} \Psi(u)}{\sum_{bs \in BS} P(bs) \times t}, \tag{15}$$

where *P(bs)* refers to base station power (*J/s*) and *t* refers to time (s). The amount of data (MB), that is transmitted to users, is divided into the sum of power consumption (*J*) of base stations.

Experience: Each replay buffer entity is called an *experience*. The experience is a combination of *state, action, reward,* and *next state* (action result). It is denoted as $e^k = (S^k, A^k, R^k, S^{k+1})$. These entities are used to train deep neural networks.

3. Experiments

In this section, the environment setup and details of experiments are explained. In order to reproduce the experiments, we provide simulation parameters and P-DDPG parameters in Sections 3.1 – 3.2. In the sections that follow the simulation model and parameters, our experiments take place. While constructing our experiments, we aim to prove that P-DDPG model can be used for energy efficiency, therefore, we ask the following questions:

- What is the motivation of developing P-DDPG algorithm? As we claim that, Vienna simulator is a slow environment, therefore, we modify DDPG algorithm to work with multiple environments to reduce the training time. Section 3.3 contains experiments of the run time of the simulation and CTI selection. This section also includes the benefits of running multiple environments with P-DDPG algorithm. To sum up, Section 3.3 contains experiments related to parameter selection and benefits of P-DDPG algorithm.
- Are we really energy efficient? We are trying to develop a model that can increase energy efficiency while maintaining QoS. To test our algorithm, we construct two scenarios with 50 user equipments (UEs) and 100 user equipments. We train our algorithm on these scenarios independently. Section 3.4 provides the details of the experiments and their results.
- Final question is that what happens if perturbation occurs in the network? Can the model still be trained? The base stations could shut down suddenly because of different reasons such as internal errors, firmware update, hardware change. We try to prove that after restarting the system, P-DDPG model can continue to manage the network in an energy efficient way. Section 3.5 gives the modelling details and experiment results.

3.1 P-DDPG Model and Simulation Environment

The P-DDPG model is developed by using TensorFlow with Python language. With socket programming, we developed a manager class to create a private channel for each environment. Thanks to the manager class, P-DDPG algorithm becomes capable of supporting distributed environment. P-DDPG algorithm is constructed according to [10]. Actor and critic networks are also referenced from there. Table 1 shows parameters of P-DDPG algorithm. In order to improve the learning stability, target networks are updated with the learning parameter of τ. This means that target network values slowly track the learned network with respect to that parameter.

Table 1 P-DDPG Parameters

Parameter	**Value**
Actor Learning Rate	10^{-4}
Critic Learning Rate	10^{-3}
γ	0.99
τ	10^{-3}
Replay Buffer size	1000000
Mini Batch Size	64
CNRC	40

3.2 Parameters of the Simulation Environment

The proposed power management algorithm is implemented for omnidirectional 7 BS with hexagonal geometry scenario to show its capabilities with different UE counts. To provide a fair evaluation we applied realistic traffic load such as video streaming which is modelled based on real-life LTE networks [20]. In this simulation, we focused on a downlink scenario where users' packet requests are varying during the runtime. The details of the SINR calculation and environment pathloss are stated in [21]. The author notes that in their formulation, the constant K at 6910 KM^{-1} corresponds to the COST Walfisch-Ikegami model for an urban environment. The pathloss simulation parameters and traffic models are summarized in Table 2.

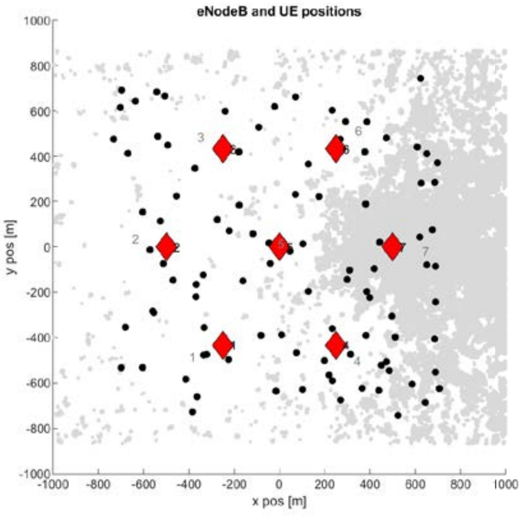


Figure 4 Example Distribution of Environment from Vienna Simulator

Vienna LTE system level simulator consists of different modules including antennas, channel models, network generation, schedulers, traffic models and etc. The proposed power management module is implemented in the network generator module to applied the expected modifications in the lowest level to make the proposed model applicable in real-life cellular networks.

In Figure 4, diamonds show base stations. Base stations are configured as omni-directional. Points refer to UEs. UEs are distributed randomly. Thanks to the random distribution, our model will try to solve generalized problem.

Table 2 Simulation Parameters

Parameter	**Value**	**Reference**
Frequency	2.14 GHz	[22]
K	6910 KM^{-1}	[21]
γ	4	[21]
N_0	$10^{-15.82}$	[21]
Subcarrier Frequency	15 kHz	[22]
Macro BSs Max Power	40 W	[22]
RRH Antenna Gain	Omni-directional	[23]
Path Loss Model	$128.1 + 37.6 \log_{10}{}^{R}$, R in km	[22]
Noise Power Spectral Density	-174 dBm/Hz	[22]
Receiver Noise Figure	9 dB	[22]
Feedback	CQI	[23]
Maximum TX power of BS	40	
Feedback Delay	3 TTI	
Scheduler	Round Robin Traffic	
Traffic Model	Video Stream	
UE speed	Stationary	
Number of Macro BS	7	
TTI	1 ms	
Communication Time Interval (CTI)	40 TTI	
Simulation Length	200 CTI	
Simulation Area	2000 m × 2000 m	
Active UEs	50, 100	

3.3 Experiment 1: Parameter Selection and P-DDPG Benefits

The environment measures the network status at each transmission time interval (TTI). Because of the realistic behaviors of users, the measurements of each TTI does not directly represent the network status. Since users are sometimes idle, sometimes actively use the network, the traffic load and average users' throughput fluctuates. Therefore, the agent cannot take decisions according to each TTI measurement. The communication time interval (CTI) is defined as a state-action exchange interval between environment and agent. To find optimum CTI, we run the simulation with different CTI. Each simulation records 200 states and we measure the standard deviation of these states in term of users' throughput.

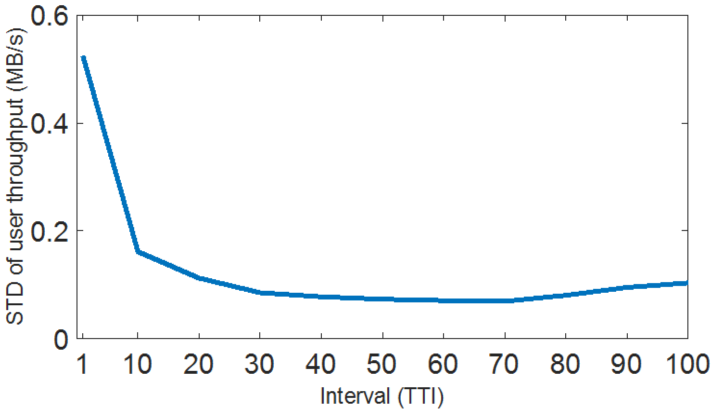


Figure 5 Standard Deviation of Average User Throughput at Different CTI Values

Figure 5 shows the effect of CTI on the standard deviation of average users' throughput. When the communication time interval increases, in term of number of TTI, the fluctuation of average users' throughput decreases. This figure proves that each transmission time interval does not represent the network status and we need to consider some time interval to state-action exchange.

The second important point of CTI selection is run time. Vienna simulator is a complex, realistic LTE system level simulator and it consists of different modules. Therefore, simulation of the real-life network in Vienna Simulator is costly and it is time required task. Run times of different CTIs are shown in

Figure 6. Each simulation runs until recording 200 states, therefore, the simulation time on a TTI basis is calculated as CTI × 200. When the simulation time increases, then the run time of environment is increases.

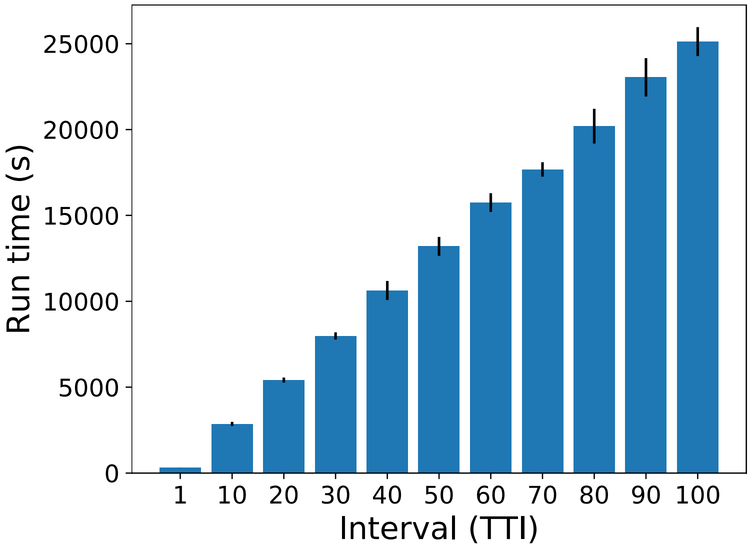


Figure 6 Analysis of Different Communication Time Interval (CTI) Run Time

DDPG algorithm can train itself after each state-action exchange. The bottleneck of training can be analyzed under two categories, one of them is agent response time and another one is environment response time. In our case, the environment's run time is significantly smaller than the agent response time, so the number of training per unit time is limited by the environment's run time. We propose P-DDPG model to increase training per unit time. The P-DDPG model can run with multiple environments in parallel.

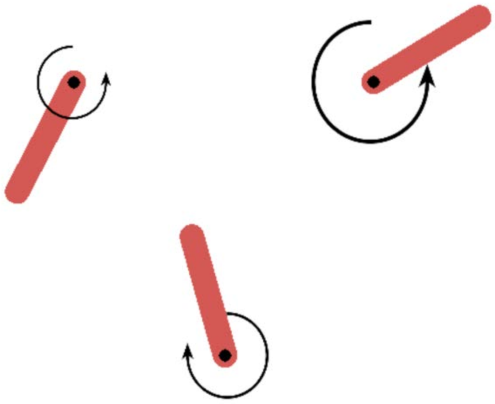


Figure 7 Snapshots of Pendulum Environment

Effect of environment response time and P-DDPG algorithm are tested on Pendulum-v0 which is one of the well-known OpenAI Gym environment. OpenAI Gym is a toolkit for developing and comparing reinforcement learning algorithms [24]. Figure 7 shows the snapshot of pendulum problem. The arrows illustrate the magnitude of the action and the direction of the action. The problem is that trying to keep a pendulum standing up by taking actions. The action is a value between -2.0 and 2.0, representing the amount of left or right force on the pendulum. Since the Vienna Simulator is a time-consuming environment as shown in Figure 6, we added a delay to the pendulum environment to observe the effect of the environment run time on learning time.

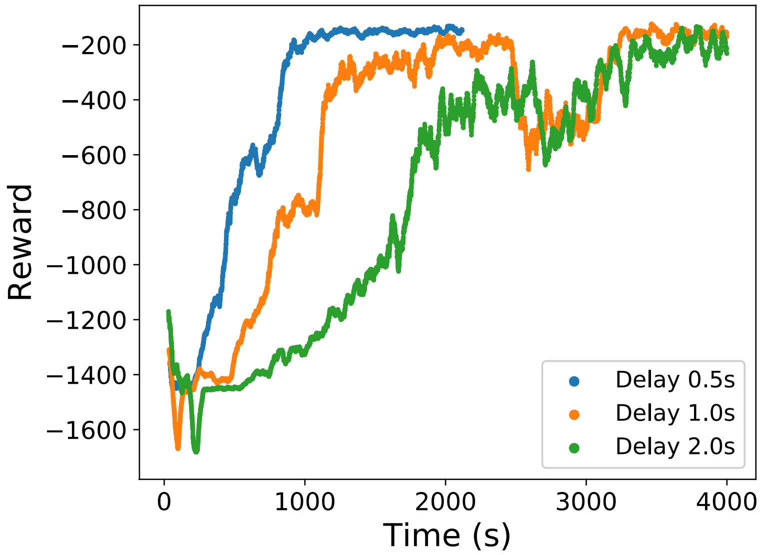
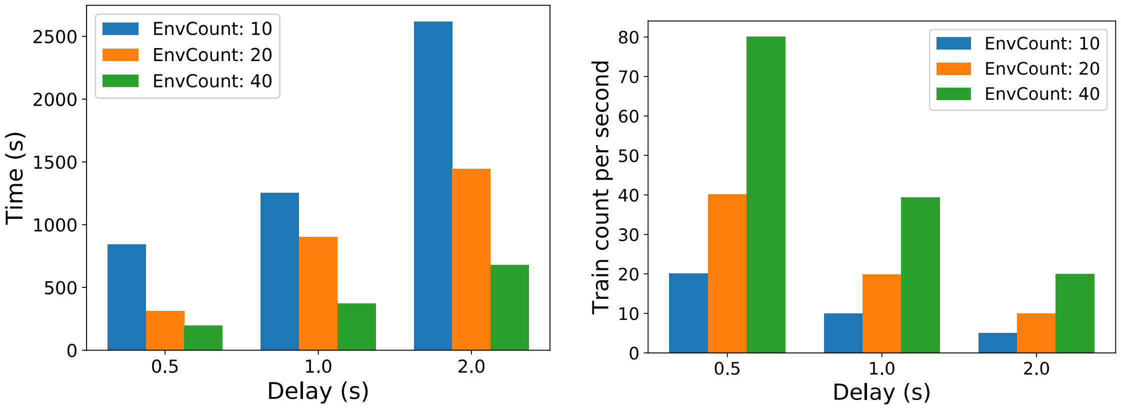


Figure 8 The Effect of Run Time of the Environment on Learning Time

Figure 8 shows environment run time effect on training time. We simulate pendulum problem on environments which have 0.5, 1- and 2-seconds delay. In these results, we run 10 multiple environments as a parallel and these results shows the moving average of last 100-episode rewards. According to convergence time of each simulation, when the environment run time increases, the learning time of the algorithm is also increased. The fastest trained model is achieved with the minimum delayed environment.



(a) Training Time

(b) Training Count per Second

Figure 9 Running Multiple Environments Effect

Running multiple environments is examined in Figure 9. We examine the effect of environment run time and the effect of the environment count on learning. The environment counts are 10, 20 and 40 where delays are 0.5, 1 and 2 seconds. We recorded run times of simulations when the average reward value of the last 100 simulations exceeded –250. Figure 9 shows that when the environment count increases, algorithm learning time decrease. For each delay value, increase in the environment count positively affects the learning time. Fastest learning time is achieved with lowest delay and highest environment count. Reinforcement learning models need trial and error. The algorithm needs to be trained as much as possible to complete learning. Hence, training count per second is important to improve algorithm. Figure 9 shows training count per second at different configurations. The delay and environment count effects on training per second are observable in this figure. There is an inverse ratio between delay and training count per second. Increase in the delay causes a decrease in the training count per second. Conversely, there is a direct correlation between the environment count and training count per second. When the environment count increases, then the training count per second increases.

To sum up, each TTI measurement does not represent the network status (Figure 5), therefore, state-action exchange has to be done at certain intervals which we called communication time interval (CTI). Vienna Simulator is a slow environment which takes time to simulate the network (Figure 6). Training is directly related with environment run time (Figure 8). Therefore, while selecting communication time interval we have to consider run time and we need to choose long enough CTI that describes the network in term of low fluctuation. We empirically choose CTI length as 40 TTI with considering these reasons. Our motivation of developing P-DDPG is the slow environment. Since the environment slow, we need to run multiple environments to increase training count per second to decrease the learning time (Figure 9).

3.4 Experiment II: Energy Efficiency under Stationary Scenario

Our problem is reducing the energy consumption of base stations while maintaining UE's QoS parameters. We try to solve this problem with our P-DDPG model. QoS parameters, that are defined in Equation 2 to 5 are basis on UEs' wideband SINR and UE's throughput. We first calculated the baseline values without running our model on the network. We run simulation 40 times for each scenario and we observe the network without taking any action. UEs and small cells are randomly distributed. The QoS distributions obtained in these observations are called baselines. We calculated acceptable QoS values (threshold values) for the network based on baseline values. These threshold values are used while training our model. We have compared baseline values with the last 40 results that pass CNRC.

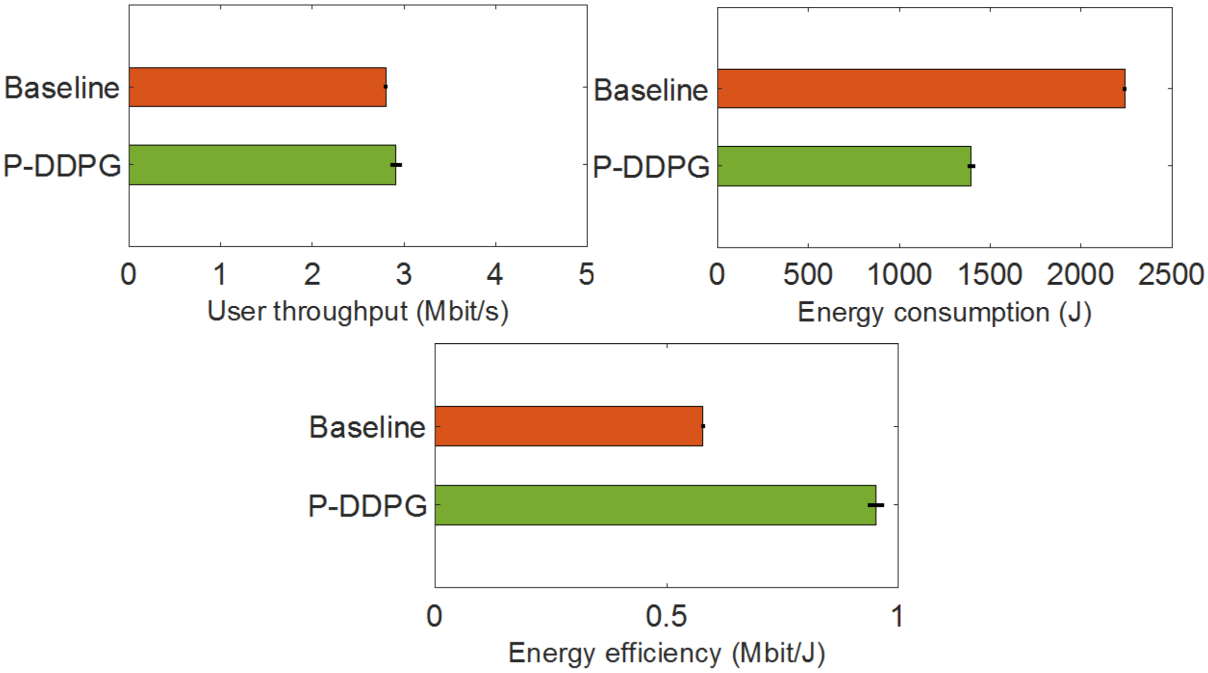
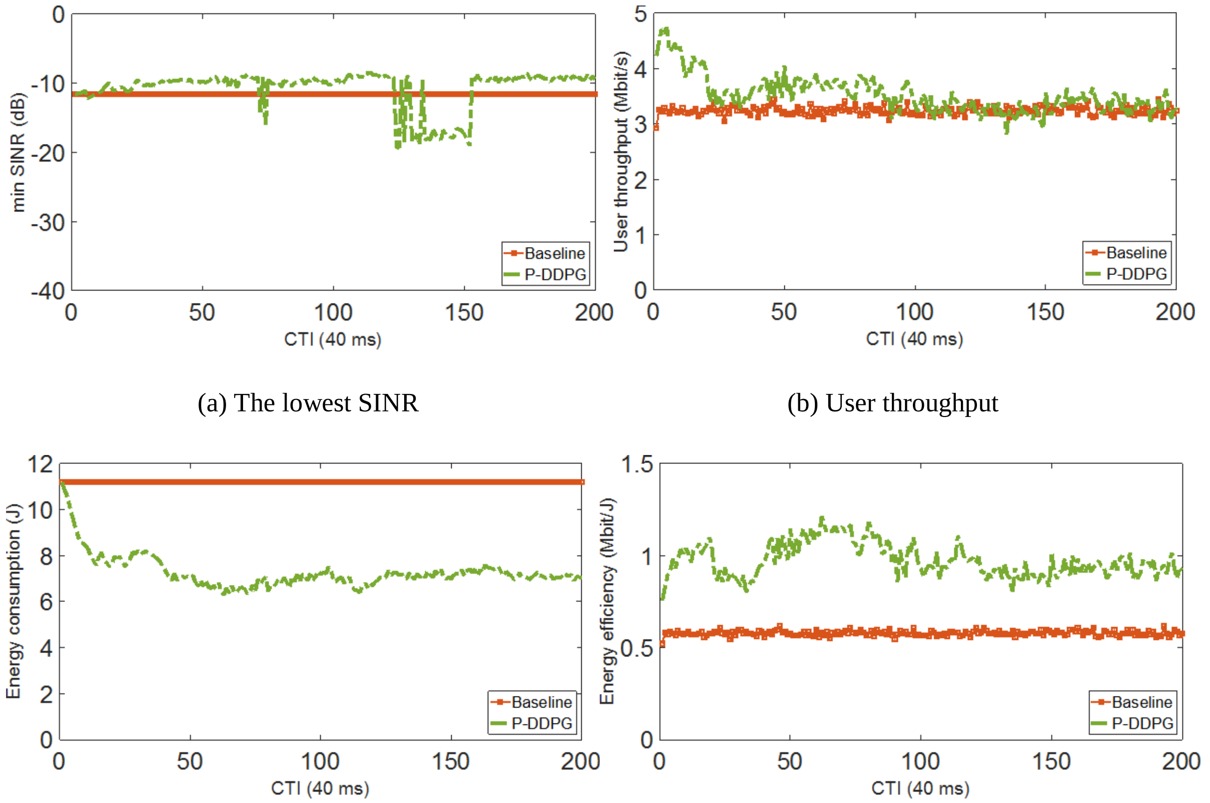


Figure 10 P-DDPG Algorithm Effects on The Environment with 50 UEs

We observed the proposed model effects in different environments. We prepare two different test scenarios to show the effect of the environmentalist energy efficient model. One of them is composed of 50 users and the other 100 users. In our observations, QoS parameters such as average throughput of users, the lowest SINR value received by users, energy consumption and energy efficiency are analyzed. The average throughput of users gives information about network usage. Users on the edge or users which are far from base stations generally get the lowest SINR value. In that case, even if we maintain the average throughput of users, some users cannot reach the network because of the poor SINR. Therefore, we compare the lowest SINR value received by users to observe the effect of the proposed model on poor users. Energy consumption and energy efficiency are the main targets that we need to improve. The realistic natural environment results of Vienna Simulator are called the Baseline, while the results of the trained model are called P-DDPG.

Figure 10 - 12 show P-DDPG effects on the environment with 50 UEs. Figure 10 is obtained by evaluating 40 simulation outputs. As we can see in Figure 10, by applying the P-DDPG model, the amount of energy consumption is reduced, while the overall UEs throughput is preserved, and in some cases, they even enhanced slightly. In sparse scenarios, we achieve up to 67% increase in energy efficiency by using the P-DDPG model in the dynamic environment.

In Figure 11, a single environment's lifetime is presented to observe the realistic environment behaviors and the P-DDPG model effects. In this work, in order to provide a real-life condition, we simulate a dynamic network where UEs have various behavior (such as the amount of received or transmit data); therefore, QoS requirements will be varied in each TTI. Even though it is challenging to maintain the QoS parameters in this variability, our model is trained to maintain QoS while increasing energy efficiency. Figure 11 shows the lowest SINR value received by users. Although there are some deviations because of the handover, the lowest SINR value is preserved. The average throughput of users is observable in Figure 11. Because of the user behaviors, there are fluctuations but the P-DDPG model successfully maintains the average throughput of users.



(a) The lowest SINR

(b) User throughput

(c) Energy consumption

(d) Energy efficiency

Figure 11 P-DDPG Algorithm Single Environment's Lifetime Effects on The Environment with 50 UEs

The P-DDPG model maintains QoS parameters while reducing energy consumption. In Figure 11, the effect of applying P-DDPG on energy consumption is presented. As it is shown, the proposed model by taking varied decisions over time can enhance energy consumption in the network for about the optimum level for the environment. The effect of fluctuations in user throughput and energy consumption on energy efficiency is also seen in Figure 11. The P-DDPG model always keeps energy efficiency higher than baseline. Due to users' behaviors, throughput is changed naturally. This explains fluctuations in energy efficiency. During a single episode, model actions which are increasing or reducing BS's powers are shown in Figure 12. The P-DDPG model plays with BS's power and find an optimum energy level for each of them.

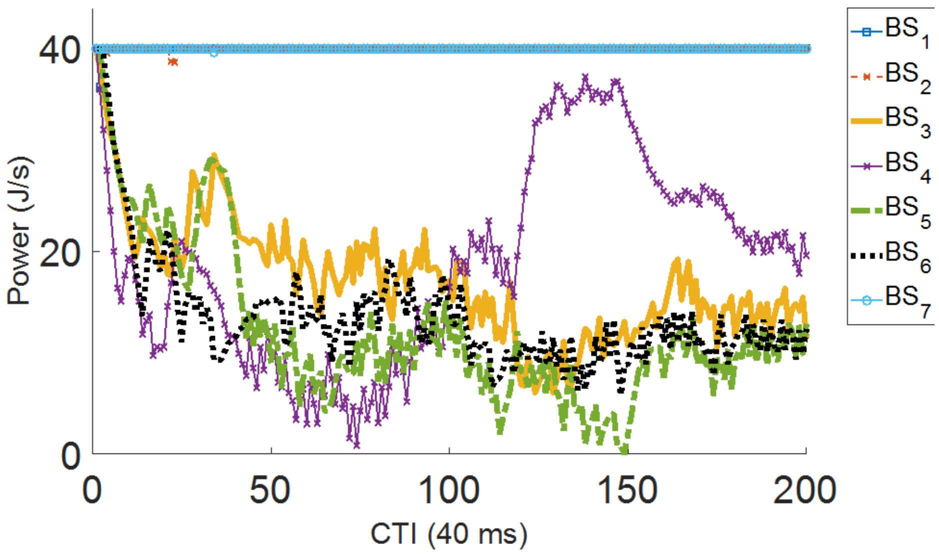


Figure 12 BS Power Changes in a Single Environment with 50 UEs

Another test scenario is constructed with 100 UEs. We illustrate the dense scenario results in Figure 13 and Figure 14. Like the sparse scenarios, the P-DDPG model can increase energy efficiency while maintaining QoS parameters. In the dense scenarios, we achieve up to 32% increase in energy efficiency. When the user density increases, users are more affected by power reduction of base stations in term of users' throughput. Therefore, the P-DDPG model maintains the power level of BSs at high and consume more power with respect to the sparse scenarios to protect the QoS. Hence, the increase in energy efficiency is less than the sparse scenarios. 40 simulations' average results are presented in Figure 13. As we can see in this figure, the P-DDPG model efficiently maintained the QoS parameters during the simulation. As it is shown, during multiple simulations, average power consumption is reduced and energy efficiency is increased when we use the P-DDPG model.

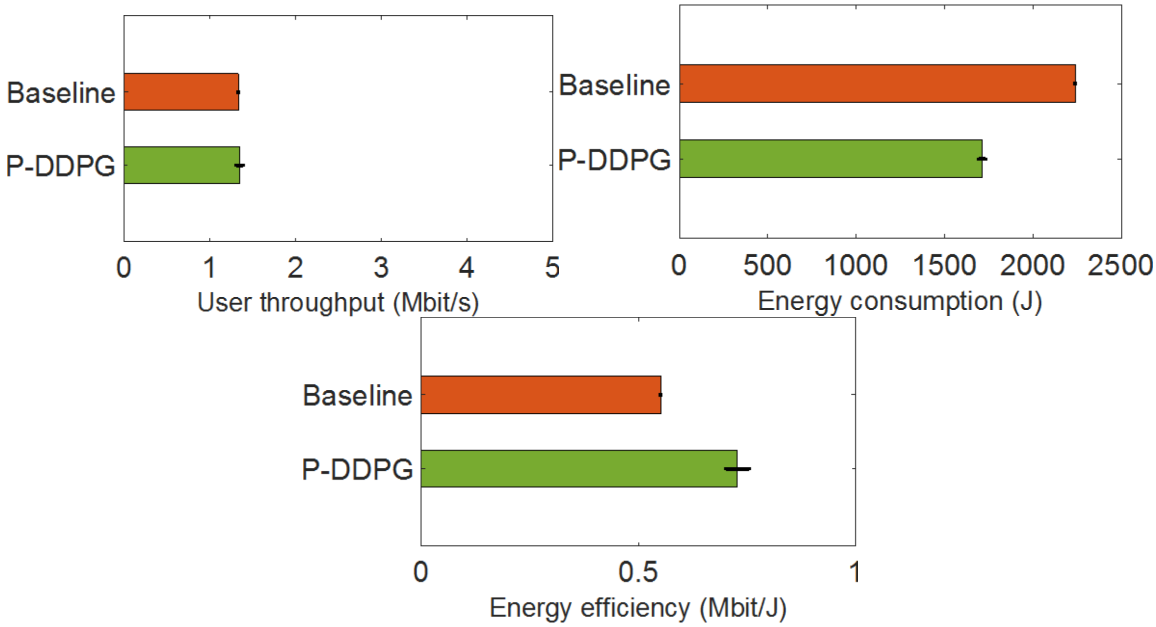
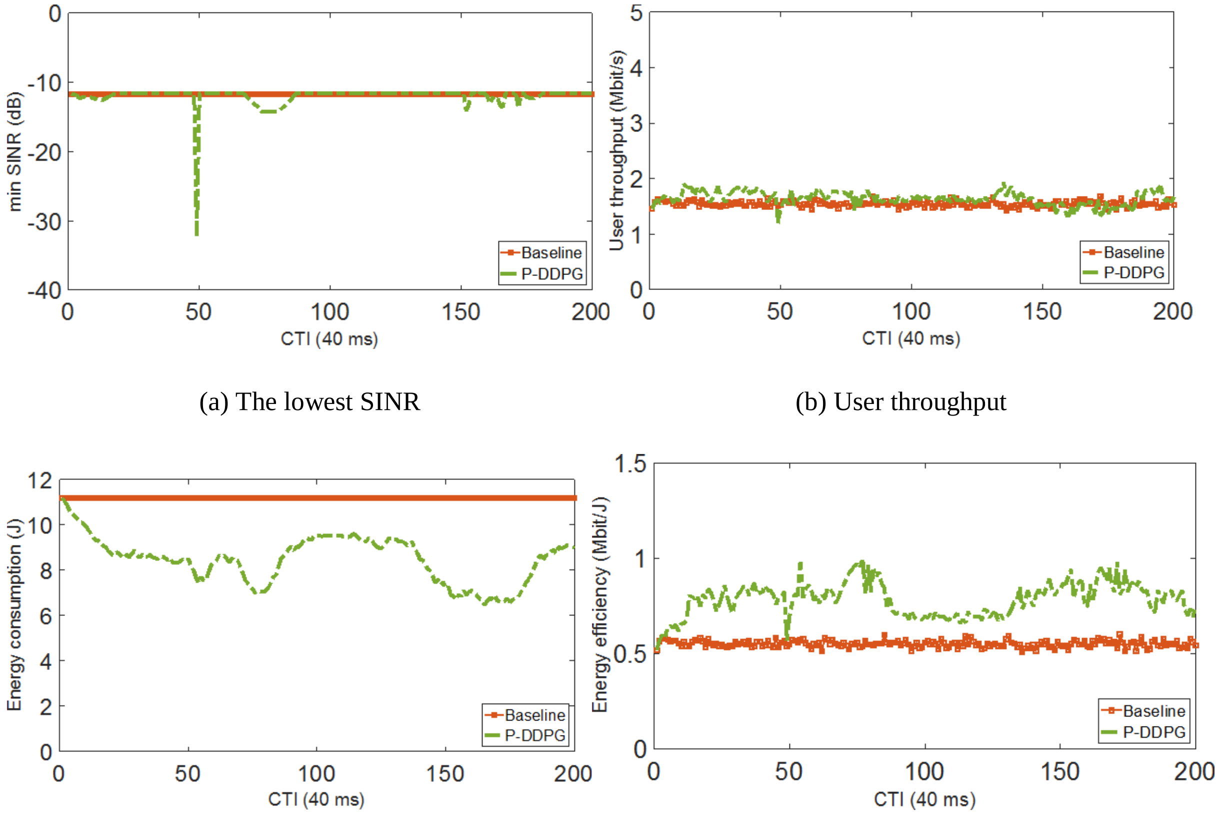


Figure 13 P-DDPG Algorithm Effects on The Environment with 100 UEs

Figure 14 shows the lowest SINR value received by users and throughput changes during one episode in the dense scenario. It appears in these figures that the P-DDPG model maintains SINR and UEs' throughput. There is a sharp reduction in the lowest SINR at 49th CTI, which may be caused handover. Except for 49th CTI, the model maintains the lowest SINR value, similar to the base. BS's power changes in the dense scenario as shown in Figure 12. The P-DDPG model acts according to the scenario requirements and keeps QoS above the threshold, and reduces energy consumption. By reducing BS's powers, it increases energy efficiency as shown in Figure 14.



(a) The lowest SINR

(b) User throughput

(c) Energy consumption

(d) Energy efficiency

Figure 14 P-DDPG Algorithm Single Environment's Lifetime Effects on The Environment with 100 UEs

In both sparse and dense cases, the P-DDPG model is trained and it decreases energy consumption. The P- DDPG model has the capability of handling natural user behavior, and it can find an optimum energy consumption level for both cases.

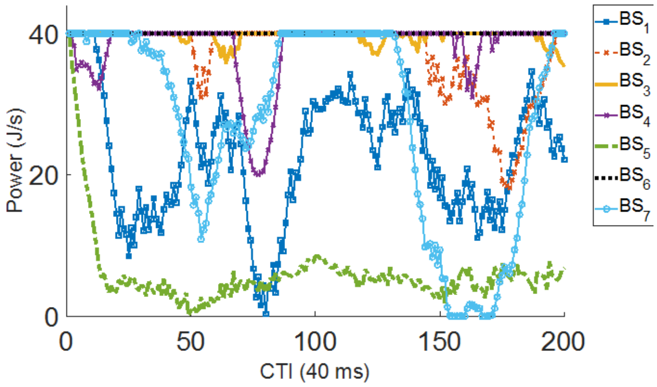


Figure 15 BS Power Changes in a Single Environment with 100 UEs

3.5 Experiment III: Energy Efficiency under Perturbation in the Network

The P-DDPG model stationary scenarios' energy efficiency results are presented in the previous section. In this section, we test the P-DDPG model when the perturbation occurs in the network. The base stations could shut down suddenly because of different reasons such as internal errors, firmware update, hardware change. Therefore, the proposed models should have a tolerance to sudden base stations turn off. To test this scenario, we set up a simulation with 50 users and 7 base stations. We set simulation length to 200 CTI. At 25th CTI, we close 3 out of 7 base stations. In Figure 16, the squares show the closed base stations and the diamonds show the rest. At 50th CTI, we restart the system and set the closed base station powers to the maximum level. During that time, model actions are neglected and we keep base station powers constant.

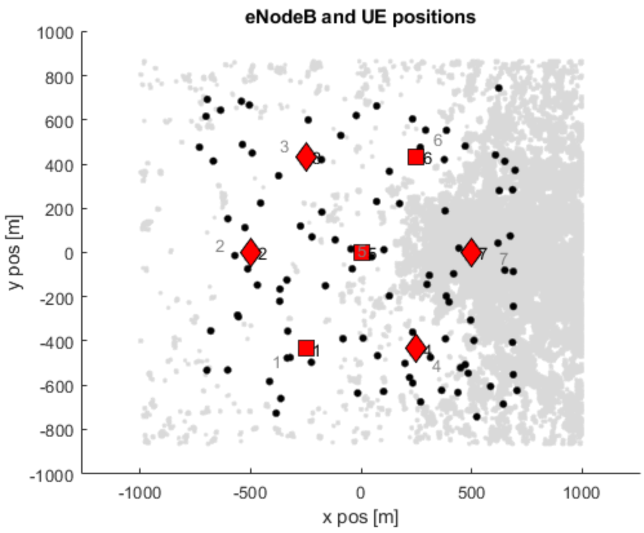


Figure 16 Perturbation Scenario Illustration

The reward function is updated for this scenario. During 25 ≤ CTI ≤ 50, the model is neither rewarded nor punished. We set R = 0 during perturbation. For the perturbation scenario, the reward function is formulated as $R^k = min(R^k_{\mathrm{OS}_2}, R^k_{EC})$, where R_{OS_2} refers to overall status of users' throughput reward (Equations 4 – 12) and R_{EC} refers to energy consumption reward (Equation 13).

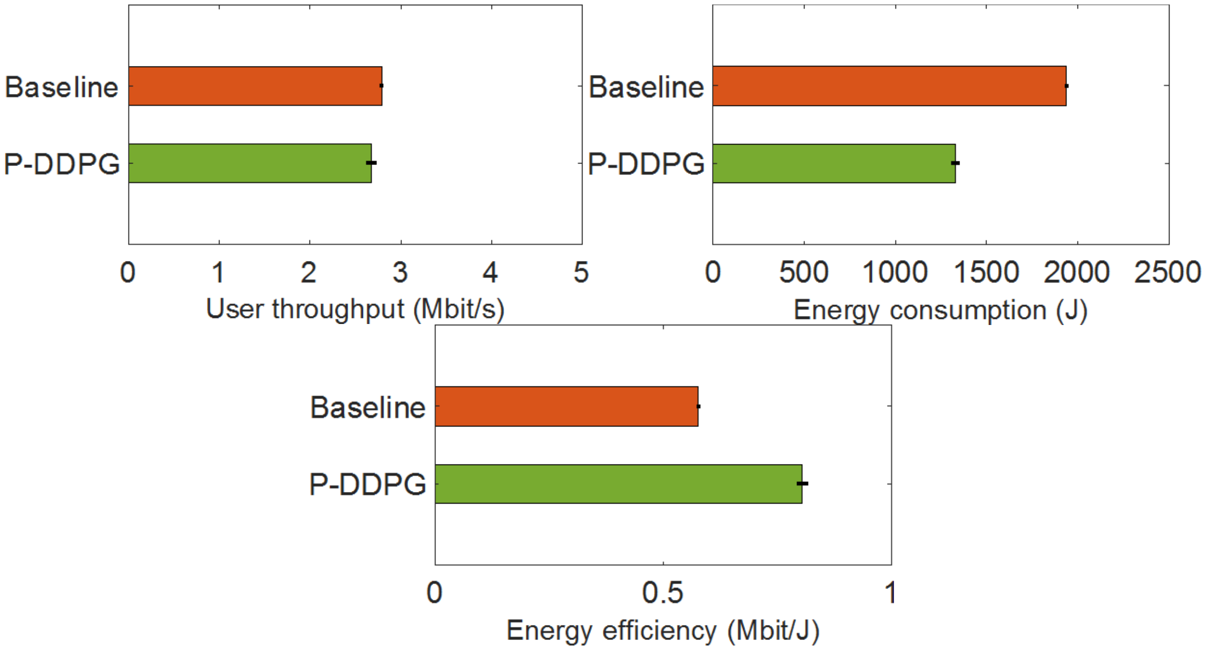
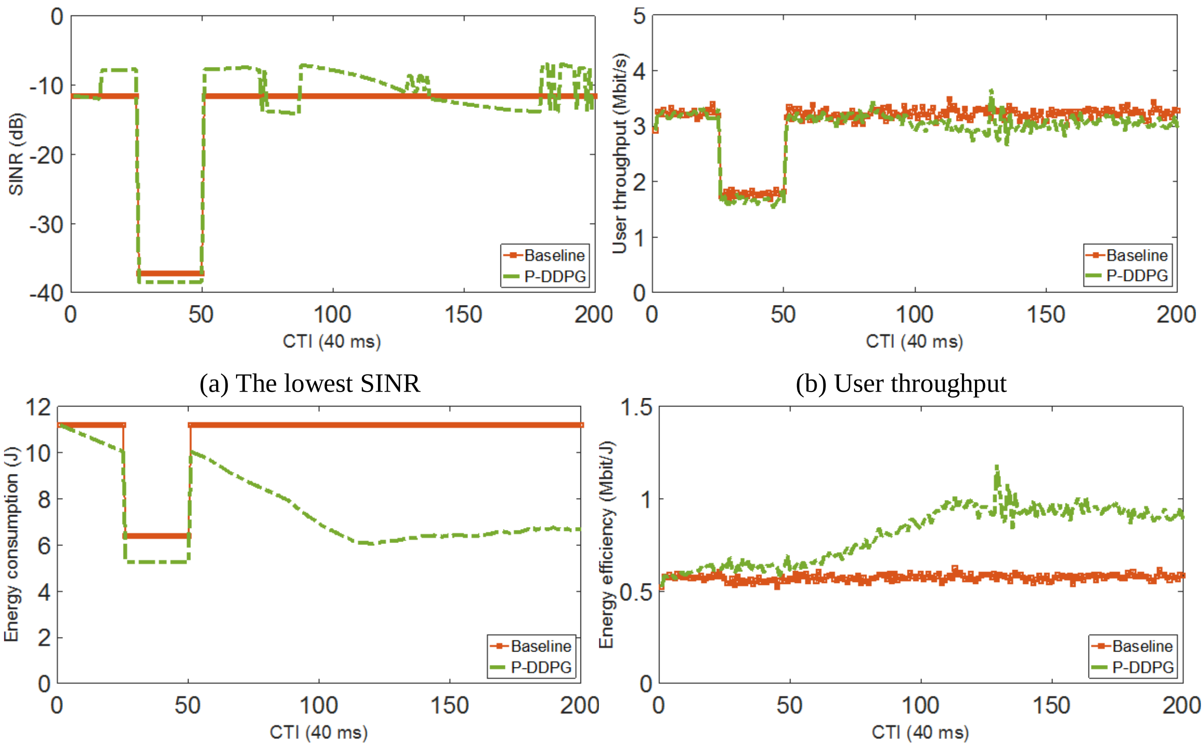


Figure 17 P-DDPG Algorithm Effects on The Environment in Perturbation Scenario

We illustrate the perturbation scenario results in Figure 17 and Figure 18. In perturbation scenario, P-DDPG model can increase energy efficiency while maintaining QoS parameters. We achieve up to 39% increase in energy efficiency.



(a) The lowest SINR (b) User throughput

(c) Energy consumption (d) Energy efficiency

Figure 18 P-DDPG Algorithm Single Environment's Lifetime Effects on The Environment in Perturbation Scenario

In this experiment, the average throughput of users and the lowest SINR value received by users are compared. The average throughput of users gives information about network usage. Hence, the average throughput of users is important with respect to maintaining QoS. QoS is not only depended on average throughput. Users on the edge or users which are far from base stations generally get the lowest SINR value. In that case, even if we maintain the average throughput of users, some users cannot reach the network because of the poor SINR. Therefore, we also try to maintain the lowest SINR value received by users to consider poor users.

Figure 17 is obtained by evaluating 40 simulations outputs. As shown in this figure, the P-DDPG model efficiently maintained the QoS parameters during the simulation like previous scenarios. The average throughput of users almost while reducing energy consumption. There is a 4% decrease in the average throughput of users where energy consumption is reduced by 30%. As a result, energy efficiency is increased by 39% by using the P-DDPG model.

In Figure 18, a single environment's lifetime is presented. The base station sudden shutdown effect is manifestly observable in these figures. There is a significant change between 25th CTI and 50th CTI. The shutting down 3 out of 7 base stations at 25th CTI sharply decrease average throughput of users. Figure 18 shows the lowest SINR value that is received from users. There is a sudden decrease in the lowest SINR value because of the shutdown.

After restarting closed base stations at 50th CTI, the P-DDPG model successfully continue to the management of the network. The average throughput of users is almost equally maintained with baseline. By using the P-DDPG model, energy consumption is significantly reduced after 50th CTI. The model tries to find optimum energy consumption level by changing base stations transmit powers. During that time, it increases energy efficiency and preserves the average throughput of users.

Effects of the sudden shutdown and the model actions on base station transmit powers are observable in Figure 19. The base stations 1, 5, and 6 are turned off during $25 \leq \text{CTI} \leq 50$. After 50th CTI, the P-DDPG model plays with BS transmit powers to find the optimum energy consumption level for each of them.

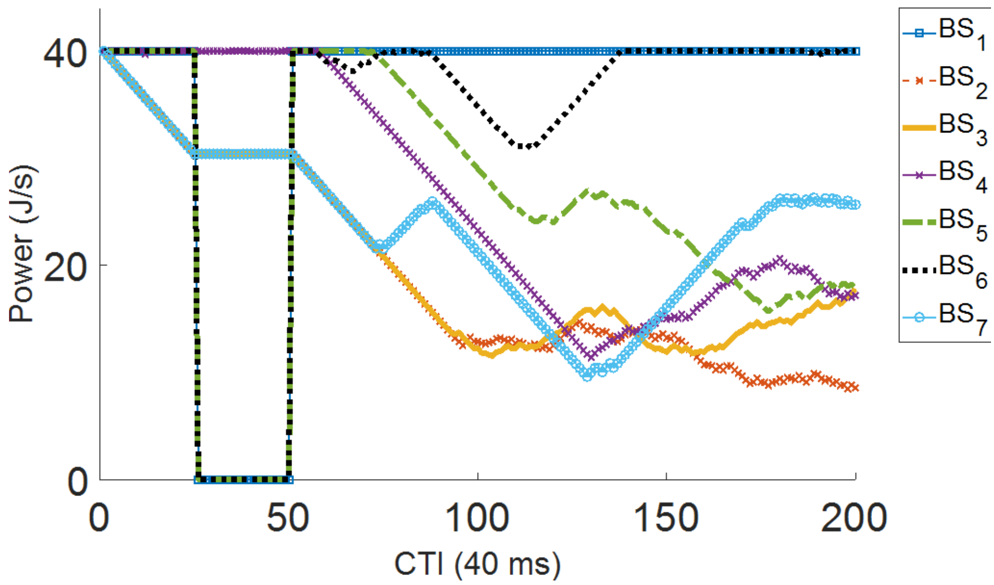


Figure 19 BS Power Changes in a Single Environment in The Perturbation Scenario

To conclude, in this experiment, the perturbation scenario is tested. The base stations could shut down suddenly because of different reasons such as internal errors, firmware update, hardware change. Users' QoS parameters adversely affected by sudden shutdowns. The P-DDPG model has the capability of continuing its management after restarting the system. It maintains users' QoS parameters while increasing energy efficiency. In perturbation scenario, we achieve up to a 39% increase in energy efficiency.

4. Related Work

Given the importance of the subject, several studies have been carried out to increase network energy efficiency using different approaches. In general, these studies try to adjust a viable sleeping strategy to achieve energy efficiency using analytic models or machine learning.

In the first group, researchers try to find an optimum sleeping strategy by modelling the network analytically. In [25], the authors aim to quantify the trade-off between energy consumption and throughput in a heterogeneous cellular network where small cell BSs have four distinct power-saving modes. These are On, Standby, Sleep and Off with power consumption ratios given as 100, 50, 15 and 0 per cent, respectively. They used a static traffic model which treats all users as stationary with known positions. Instead of using discrete power levels, our proposed model can use continuous power levels. It increases complexity but thanks to the machine learning, the proposed model has the capability of handling continuous power level adaptation. Feng et al. list new challenges of a design of BSs sleeping strategy in 5G networks [7]. They provide a comprehensive review of recent advances on ON-OFF switching mechanisms in different application scenarios. They list various ON-OFF switching problems and known solutions. Their claim is that ON-OFF scheduling is generally an NP-hard problem and solving with standard techniques is unfeasible. Moreover, they point the application of machine learning techniques on the network as future research. Cai et al. propose to dynamically change the operating states of the BSs (as on and off) to reduce the power consumption of the heterogeneous networks (HetNets) [26]. They consider location and user density-based operation scheme to optimize power consumption. These studies describe the network by using mathematical models. After modelling, they try to solve an optimization problem to find an optimum sleeping mode for each BS. Mathematical models try to find threshold values such as the number of users that are assigned to each BS, or throughput threshold. Threshold values are used for sleeping decisions. When we consider the dynamic nature of the system, these models have to make very restricting assumptions for a network to perform well. Unlike machine learning used models, modelling the stochasticity is a challenging problem as well as solving it.

On the other side, with the advent of software defined networking, machine learning approaches are available for a network. Historical data and online learning concepts are key-enablers for using machine learning in the network management. Reinforcement learning is well suited to the online and continuous nature of the network management problem. In [27], Lu et al. develop a RL model for cellular networks with coordinated multipoint communication. The model aims to find an optimum solution for ON-OFF switching. They use the Q-learning algorithm while modelling solution. Their main focus is macro base stations and they use only ON-OFF as an action. Therefore, they cannot use the intermediate power values of the base stations. Sharma et al. use an actor-critic reinforcement learning approach for ON-OFF switching in HetNets [28]. They emphasis transfer learning benefits and they point the relation between energy efficiency and delay importance. RL is also used for energy efficient resource allocation in 5G heterogeneous cloud radio access networks by Al Qerm and Shihada [29]. They build an online Q-learning model for resource allocation. Their action and state space are relatively larger than previous studies, however, because of the Q-learning, the curse of dimensionality problem is there for them. Ghadimi et al. try to use RL for transmit power adaptation [30]. Their action space is {0, ±1, ±3}. Limited action space and state space are the known phenomenon of Q-learning. There are studies for developing RL models to optimize energy efficiency in small cells such as Wi-Fi routers, 4G home eNBs and 5G home gNBs [31]. Researchers point the problem of small cell energy consumption in next generation networks in their study. They try to optimize energy consumption by considering QoS. They transform continuous decision variables into discrete ones to reduce complexity and to fit their models which are based on regret learning based RL and fictitious play based RL. In contrast to their work, we focused macro cells and continuous actions are supported in our proposed model.

Moreover, reinforcement learning is used for various control problems in the mobile network. In [32], Wang et al. use federated deep reinforcement learning for decentralized cooperative edge caching. They address to content replacement problem by modeling as a Markov decision process and they use a double deep Q-network (DDQN) to find a solution in the noncontinuous huge space. The time division duplexing control problem is also adressed by using reinforcement learning. In [33], authors work on duplexing framework that let network to dynamically adapt according to the traffic demands. At the same time, intercell interference is mitigated by using their suggested approach. They model the problem by using DDQN.

Unlike previous works, we used deep deterministic policy gradient-based reinforcement learning. Our proposed model is constructed to support continuous state space and continuous action space. This is a novel approach that finds the optimum power consumption in the network by using deep deterministic policy gradient-based reinforcement learning that trained in a realistic environment.

5. Conclusion

In the next-generation networks, due to the increase in the BS density, environmentalist approach will be essential. Autonomous systems will be more involved in the next generation network. These autonomous systems should be trained in a simulation environment which is close to real life. In this paper, we present a dynamic machine learning based energy saving model. P-DDPG is an extended version of DDPG which has the capability of work with parallel environments. Unlike previous works, we train our model on Vienna Simulator, which is known as realistic and advanced network simulator. Our experiments show that it is possible to achieve up to 32% increase in the energy efficiency in a dense scenario and up to 67% increase in sparser scenarios. P-DDPG successfully manages networks in term of reducing energy consumption and maintaining QoS. As future work, the proposed approach can be tested in environments with mobile users. By taking advantage of transfer learning, it is possible to work in environments with mobile users as a continuation of this study. A fully autonomous system running on the network can be obtainable by using P-DDPG models that are trained for each cluster.

Acknowledgments

This study is supported by Turkcell under BTK Graduate Scholarship Program.

List of Abbreviations

BS	Base Station
CNRC	Consecutive Negative Reward Check
CTI	Communication Time Interval
DDPG	Deep Deterministic Policy Gradient
DDQN	Double Deep Q Network
DQN	Deep Q Network
HetNet	Heterogeneous Networks
LTE	Long-Term Evolution
MEC	Multi-access Edge Cloud
ML	Machine Learning
NFV	Network Function Virtualization
OS	Overall Status
P-DDPG	Parallel Deep Deterministic Policy Gradient
QoS	Quality of service
RAN	Radio Access Network
RL	Reinforcement Learning
SINR	Signal to Interference Ratio
SN	Software networks
TTI	Transmission Time Interval
UE	User Equipment

References

[1] A. Usman, I. Ozturk, A. Hassan, S. M. Zafar and S. Ullah, "The effect of ICT on energy consumption and economic growth in South Asian economies: an empirical analysis," *Telematics and Informatics,* vol. 58, p. 101537, 2021.

[2] A. Abrol and R. K. Jha, "Power Optimization in 5G Networks: A Step Towards GrEEn Communication," *IEEE Access,* vol. 4, pp. 1355-1374, 4 2016.

[3] E. C. Strinati and L. Herault, "Holistic approach for future energy efficient cellular networks," *Elektrotechnik und Informationstechnik,* vol. 127, p. 314–320, 11 2010.

[4] S. Zhou, J. Gong, Z. Yang, Z. Niu, P. Yang and D. Corporation, "Green mobile access network with dynamic base station energy saving," *ACM MobiCom,* vol. 9, p. 10–12, 1 2009.

[5] M. Ismail, W. Zhuang, E. Serpedin and K. Qaraqe, "A survey on green mobile networking: From the perspectives of network operators and mobile users," *IEEE Communications Surveys and Tutorials,* vol. 17, pp. 1535-1556, 2015.

[6] M. Aykut Yigitel, O. D. Incel and C. Ersoy, "Dynamic BS Topology Management for Green Next Generation HetNets: An Urban Case Study," *IEEE Journal on Selected Areas in Communications,* vol. 34, p. 3482–3498, 12 2016.

[7] M. Feng, S. Mao and T. Jiang, "Base Station ON-OFF Switching in 5G Wireless Systems: Approaches and Challenges," *IEEE Wireless Communications,* vol. 24, p. 46–54, 8 2017.

[8] B. B. Post and H. van den Berg, "A self-organizing base station sleeping and user association strategy for dense cellular networks," *Wireless Networks*, vol. 27, no. 1, pp. 307-322, 2021.

[9] A. Al-Quzweeni, A. Lawey, T. El-Gorashi and J. M. H. Elmirghani, "A framework for energy efficient NFV in 5G networks," in *2016 18th International Conference on Transparent Optical Networks (ICTON)*, 2016.

[10] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, "Continuous control with deep reinforcement learning," in *4th International Conference on Learning Representations*, 2016.

[11] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen and L. Hanzo, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wireless Communications*, vol. 24, pp. 98-105, 4 2017.

[12] T. E. Bogale, X. Wang and L. B. Le, "Machine Intelligence Techniques for Next-Generation Context-Aware Wireless Networks," *ITU Journal: ICT Discoveries, Special Issue*, p. 1–11, 2 2018.

[13] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu and F. Kojima, "Big Data Analytics, Machine Learning and Artificial Intelligence in Next-Generation Wireless Networks," *IEEE access*, vol. 6, p. 32328–32338, 5 2018.

[14] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, p. 9–44, 8 1988.

[15] L. Raju, R. S. Milton, S. Suresh and S. Sankar, "Reinforcement learning in adaptive control of power system generation," *Procedia Computer Science*, vol. 46, p. 202–209, 12 2015.

[16] C. J. C. H. Watkins, "Learning from Delayed Rewards," Cambridge, 1989.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," in *Neural Information Processing Systems (NIPS) Workshop on Deep Learning*, 2013.

[18] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra and M. Riedmiller, "Deterministic Policy Gradient Algorithms," *31st International Conference on Machine Learning, ICML 2014*, vol. 1, 6 2014.

[19] M. Rupp, S. Schwarz and M. Taranetz, The Vienna LTE-Advanced Simulators: Up and Downlink, Link and System Level Simulation, 1 ed., Springer Singapore, 2016.

[20] G. Karagiannis, G. T. Pham, A. D. Nguyen, G. J. Heijenk, B. R. Haverkort and F. Campfens, "Performance of LTE for Smart Grid Communications," in *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, Springer International Publishing, 2014, p. 225–239.

[21] H. P. Keeler, B. Blaszczyszyn and M. K. Karray, "SINR-based k-coverage probability in cellular networks with arbitrary shadowing," in *2013 IEEE International Symposium on Information Theory*, 2013.

[22] 3rd Generation Partnership Project (3GPP), "Evolved universal terrestrial radio access (EUTRA)," in 3rdGenerationPartnership Project (3GPP), 2014.

[23] B. Clerckx, G. Kim and S. Kim, "MU-MIMO with Channel Statistics-Based Codebooks in Spatially Correlated Channels," in *IEEE Global Telecommunications Conference*, 2008.

[24] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba, *OpenAI Gym*, 2016.

[25] C. Liu, B. Natarajan and H. Xia, "Small Cell Base Station Sleep Strategies for Energy Efficiency," *IEEE Transactions on Vehicular Technology*, vol. 65, p. 1652–1661, 3 2016.

[26] S. Cai, Y. Che, L. Duan, J. Wang, S. Zhou and R. Zhang, "Green 5G Heterogeneous Networks Through Dynamic Small-Cell Operation," *IEEE Journal on Selected Areas in Communications*, vol. 34, pp. 1103-1115, 5 2016.

[27] H. Lu, B. Hu, Z. Ma and S. Wen, "Reinforcement learning optimization for energy-efficient cellular networks with coordinated multipoint communications," *Mathematical Problems in Engineering,* vol. 2014, pp. 1-9, 7 2014.

[28] S. Sharma, S. J. Darak and A. Srivastava, "Energy saving in heterogeneous cellular network via transfer reinforcement learning based policy," in *9th International Conference on Communication Systems and Networks (COMSNETS),* 2017.

[29] I. AlQerm and B. Shihada, "Enhanced machine learning scheme for energy efficient resource allocation in 5G heterogeneous cloud radio access networks," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC),* 2017.

[30] E. Ghadimi, F. D. Calabrese, G. Peters and P. Soldati, "A reinforcement learning approach to power control and rate adaptation in cellular networks," in *2017 IEEE International Conference on Communications (ICC),* 2017.

[31] Y. Wang, X. Dai, J. M. Wang and B. Bensaou, "A Reinforcement Learning Approach to Energy Efficiency and QoS in 5G Wireless Networks," *IEEE Journal on Selected Areas in Communications,* vol. 37, pp. 1413-1423, 6 2019.

[32] X. W. Wang, L. Chenyang, L. Xiuhua, T. Victor and Tarik, "Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching," *IEEE Internet of Things Journal,* vol. 7, no. 10, pp. 9441-9455, 2020.

[33] N.-N. N. Dao and C. S. Wonjong, "Deep Reinforcement Learning-Based Hierarchical Time Division Duplexing Control for Dense Wireless and Mobile Networks," *IEEE Transactions on Wireless Communications,* 2021.

SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 4, NO. 2, AUGUST 2021

Research Article

A Curve Fitting Modelling Approach to Forecast Long-Term Electrical Energy Consumption: Case Study of Turkey

Abdal Kasule[1], Şaban Selim Şeker[2], Kürşat Ayan[3]
[1]Corresponding Author; Department of Computer Science and Information Technology, Islamic University in Uganda, Mbale, Uganda; abdal@iuiu.ac.ug; +256 77 264 05 78;
[2]Department of Electrical and Electronics Engineering, Uskudar University, Istanbul, Turkey; selim.seker@uskudar.edu.tr
[3]Department of Electrical and Electronics Engineering, Istanbul Medeniyet University, Istanbul, Turkey; kursat.ayan@medeniyet.edu.tr

Received 17 June 2021; Revised 01 August 2021; Accepted 22 August 2021; Published online 31 August 2021

Abstract

For Turkey to achieve the targets of Vision 2023 of being in the top ten economies of the world, the eleventh National Development Plan (NDP11) focuses on ensuring uninterrupted, high-quality, sustainable, reliable and affordable energy supply. In this regard medium- and long-term energy supply-demand planning is regarded as a key input to the planning process. Medium and long-term planning is possible only when reliable forecasts are available. Using Turkey's electrical energy consumption data from 1970 to 2015, this study presents novel Gaussian, Fourier and Exponential curve fitting and extrapolation approaches to forecast Turkey's electrical energy consumption up to the year 2025. Major interest is put on how the model forecasts electrical energy consumption for year 2023 because this year marks a century of the establishment of the Republic of Turkey and all strategic plans are focused on how to achieve the targets as outline in Vision 2023. We evaluate the performance of the models on how best they forecast electrical energy consumption for the year 2023. Our forecasts for the year 2023 are 352.7TWh, 377.4 TWh, and 460.1TWh for the Gaussian, Fourier and Exponential models respectively which compare well with NDP11's estimated 375.8 TWh electrical energy consumption in 2023.

Keywords: : energy consumption forecasting, curve fitting, Vision 2023, Turkey

1. Introduction

In the late 20th century, Turkey experienced a dramatic increase in energy consumption. As expected, the energy consumption will continue to rise in the future. Energy is an enabler to the creation and generation of wealth and has great significance in economic development. Thus, for every country, energy is a vital resource. To achieve sustainable development there must be a sustainable supply of energy. Over the last decade, forecasting has become a tool used by many business corporations and governments not only to make better decisions but also gain competitive advantage. Turkey's current energy generation does not meet the existing energy demands that are raising annually by 4-6 percent until 2023 [1]. The operational electric energy sources are composed of coal, liquid fuels, electricity, natural gas, and renewable. Of this coal contributes 37.2%, liquid fuels 0.1%, natural gas 30.3%, Hydroelectricity 19.7% and renewable energy and wastes 12.7% [2]. Of the electricity generated from the above sources, the industrial sector is the biggest consumer with 45.6%, followed by household/domestic sector with 21.1%, commercial 20.4%, others 6.5%, government 4.6% and illumination 1.8% [3]. In regard to energy, the Eleventh National Development Plan (NDP11) focuses on ensuring uninterrupted, high-quality, sustainable, reliable and affordable energy supply. NDP11 further mentions that medium- and long-term energy supply-demand planning will be made. Forecasting energy

To Cite This Article: A. Kasule, S. S. Seker, K. Ayan, "A Curve Fitting Modelling Approach to Forecast Long-Term Electrical Energy Consumption: Case Study of Turkey," Sakarya University Journal of Computer and Information Sciences, vol. 4, no. 2, pp. 266–276, 2021. doi: 10.35377/saucis.04.02.953902

demand is key to supply-demand planning. To show the importance of forecasting, in NDP11 it is estimated that electricity demand will reach 375.8 TWh by the year 2023 [4]. All of the goals that contribute to achieving the high-level national objective of Vision 2023, whose objective is to make Turkey one of the ten greatest economic powers in the world by the year 2023 require a substantial amount of electrical energy supply in the country. Various studies have used different approaches to forecast Turkey's electrical energy demand and consumption. For example, using Artificial Neural Networks (ANN) Yunus and Mo [5] forecasted Turkey's sectoral electrical energy consumption until the year 2023. Gülsüm [6] forecasted regional electrical energy demand using time series, panel data and spatial panel data models. Şule et al [7] used multiple linear regression (MLR) to select independent variables from population, Gross Domestic Product (GDP), exports, imports, employment and natural gas to forecast Turkey's electrical energy consumption for the years 2015-2023 using ANN. The results show that Turkey's electrical energy consumption will vary between 337087.4 and 385006.6 Gwh by 2023. In another study, Aydin [8] used regression analysis taking population and gross domestic product as independent variables to forecast Turkey's primary energy consumption for the years 2010 up to 2025. The forecasts showed that Turkey's future energy consumption would be between 174.65 and 203.13 Million Tons of Oil Equivalent (Mtoe) in the year 2025. Turkey's Ministry of Energy and Natural Resources (MENR) has extensively used the Model for the Analysis of Energy Demand (MAED) to forecast energy demand. For comparison purposes with MAED results, MENR used the "old technique" developed in the mid-20th century to forecast energy demand. This technique simply states that every year energy demand increases by 7% of the previous year's energy consumption. All the previous studies mentioned above use either statistical and computational intelligence forecasting approaches. In either case, both approaches use various independent variables mostly social economic factors as inputs to forecast electrical energy consumption. Therefore, future values of the independent variables have to be obtained, mainly through forecasting so that electric energy can be forecasted. This means that the accuracy of electric energy forecasts depend on the accuracy the forecasted independent variables. If the independent variables are poorly forecasted, the electric energy forecasts too are going to be poor. Thus, in this study we present a novel Gaussian, Fourier and Exponential curve fitting and extrapolation approaches to forecast Turkey's electrical energy consumption up to the year 2025. The Least Squares Algorithm was used to find optimal parameters of the models. As seen in all the previous studies, we also compare results with the "old technique" used by MENR to forecast energy demand. In section 2 we present related work while section 3 discusses the methods and materials. Section 4 presents the results and we make a conclusion in section 5.

2. Related work

Electrical energy forecasting is fundamentally important to many players in the generation, transmission and distribution of energy. These include among others governments, energy suppliers, and participants in energy markets. Energy forecasts are divided into short-term forecasting, medium-term forecasting and long-term forecasting. Short-term forecasting is from one hour to one week. The forecast results of short-term forecasting are mainly useful for load balancing and pricing. Medium-term forecasting is used to forecast energy from one week to one year while long-term forecasting is for longer periods ranging from one year to up to thirty years or more. The long-term forecasting results are useful for capacity investment decisions, planning for expansion, revenue analysis and corporate budgeting,[9]. Kuster et al.,[10] in their review of electrical load forecasting models notes that despite their simplicity, regression methods are still in common use for long-term and very long-term forecasting. In contrast to regression, other statistical methods such as time series analysis are also used. Besides statistical methods, computational intelligence methods have found rich applicability in forecasting. Commonly used methods include Support Vector Machines (SVM), and artificial neural networks which are mostly used for short and very short-term forecasting. All methods take some form of historical data is used. In Barran et al.,[11] Random Forest Regression (RFR), Gradient Boosted Regression (GBR) and Extreme Gradient Boosting (XGB) were used to study wind energy prediction and solar radiation globally and locally. Their experiments showed that predictions using Support Vector Regression (SVR) for individual wind farms is greatly improved by these ensemble methods. [12] proposes a novel approach that is more accurate to forecast electricity load using Recurrent Extreme Learning Machine (RELM). The study adapts a method for training single hidden layer feed forward neural network to train the Jordan recurrent network. A comparative analysis of results with linear regression, traditional ELM, generalized regression neural network and other commonly used machine learning methods showed that RELM had achieved tremendous success in electricity load forecasting. Jinliang et al.,[13] asserts that much as short term electricity load forecasting is important for market participants, it is to a great extent influenced by both natural and social factors which make it more challenging. They thus proposed a hybrid model based on Autoregressive Integrated Moving Average (ARIMA), Wavelet Neural Network (WNN) and Improved Empirical Mode Decomposition (IEMD). To optimize the model, the fruit fly optimization algorithm (FOA) is used. The optimized model is then used to forecast short term electricity loads. Comparison of results of proposed model with other models showed that the hybrid model performed better in electricity load forecasting. SVR with a radial basis function (SVR-RBF), a multilayer feed-forward neural network (MLFFNN) and a Particle Swarm optimized adaptive neuro-fuzzy inference system (ANFIS-PSO) are implemented in,[14] to predict the speed and direction of wind and the resulting power output of a wind turbine. Using statistical indices to compare the observed and predicted values showed that the SVR-RBF model was better than the MLFFNN and ANFIS-PSO models. Unler, [15] proposed a model to forecast energy demand of Turkey based on particle swarm optimization (PSOEDF). The model takes population, GDP, imports and exports of Turkey were used as independent variables. Kankal et. al,[16] studied how a teaching-learning-based optimized artificial neural network (ANN-TLBO) performs when modelling Turkey’s electric energy demand (EED). This model uses population, imports, exports and GDP as independent variables. Back-propagation optimized ANN (ANN-BP) and Artificial Bee Colony algorithm optimized ANN (ANN-ABC) were compared with ANN-TLBO. The results showed that the ANN-TLBO models performed better than the ANN-ABC and ANN-BP models to estimate EED. An analysis of the changes in electricity generation policies by revisiting the dynamics Turkey’s electricity generation resources is done in,[17]. A Multiple Attribute Decision Making (MADM) based on an integrated method composed of Monte Carlo simulation, Borda count and entropy were used to evaluate solar photovoltaic (PV), hydro, wind, biomass, coal, nuclear and oil alternatives. The evaluation results showed that the primary sources for generating electrical energy in Turkey should be mainly composed of sources that are renewable. Kaytez,[18] proposed a model that hybridizes least-square SVM and ARIMA to forecast net electrical energy consumption for Turkey until 2022. The study results show that the hybridized SVM-ARIMA generates more reliable and realistic forecasts. Çevik et al.,[19] in their study used ANN and Particle Swarm Optimization (PSO) techniques to forecast electricity load for 24 hours of next day. ANN weights of ANN were updated in the learning phase by PSO. ANN inputs were past values of consumed electrical load data, season, and daily average air temperature data. Turkey’s monthly electrical demand between 2015 and 2018 were predicted by modelling seasonality and trend effects in [20]. They developed four different ANN models out of which the best model was selected. The best ANN model was compared with Seasonal Autoregressive Integrated Moving Average (SARIMA) model. Çeribaşı, and Çalışkan [21] estimated prospective long-term and short-term energy that can be generated by Adasu Regulator and Hydroelectric Energy Plant and Pamukova Hydroelectric Energy Plant in Sakarya Basin of Turkey. To make short-term estimates, ANN was used while long term estimates were obtained using the Innovative Sen Method. Farahat and Talaat,[22] presented a Curve Fitting prediction approach to forecast short-term load in which the optimal parameters of Gaussian model are obtained using a genetic algorithm that takes the error between actual and forecasted load as the cost function. In [23], linear regression and polynomial curve fitting are used to forecast wind and solar power production. Results showed that the R-squared and adjusted R-squared values obtained from the polynomial curve fitting model were better than those of the regression model, hence concluding that the polynomial curve fitting model is a better model.

3. Methods and materials

Electricity is the most commonly used form of secondary energy; therefore, we have developed models for electricity consumption for Turkey. In this study we develop four models to forecast electricity demand for Turkey and compare our results with the old technique used by MENR. First, we describe the "old technique" used by MENR.

3.1. The Old Technique

In the middle of 20th century, a very basic and simple technique was used in order to predict energy demand. This technique simply states that every year energy demand increases by 7% of the previous year’s energy consumption, “as shown in Equation 1”.

$$a_t = a_0 * (1.07)^{t-t_0} TWh \tag{1}$$

where a_t is the consumption for year t, a_0 is the known consumption of year t_0. Using this technique, if we take the consumption of 263TWh for year 2015, we find that the consumption for the year 2023 is 458.88TWh, “as shown in Equation 2”.

$$a_{2023} = 263 * (1.07)^{2023-2015} = 458.88\,TWh \tag{2}$$

3.2. The proposed approach

In this study we propose a novel approach of curve fitting to long-term electrical energy forecasting. Curve fitting is involves finding an appropriate or optimal function that can be used to fit a model to a given dataset. It is not a necessary requirement that the function has to pass through all of the points, the focus is to model the data with the minimum possible error between the observed data points and the fitted curve. Curve fitting is done in two phases, in the first phase a functional relation involving undetermined parameters is selected and in the second phase best estimates of the values of the parameters is made. In curve fitting, we examine whether one or more predictors (independent variables) are related to a response variable (dependent variable). The goal is to define a "best fitting model" that describes the relationship. In our study, the year was taken as the independent variable and historical load was taken as the dependent variable. The curve fitting procedure finds the specific coefficients which make that function match given dataset as closely as possible. Any type of function can be used for curve fitting. The curve fitting procedure is shown in the Fig. 1 below.

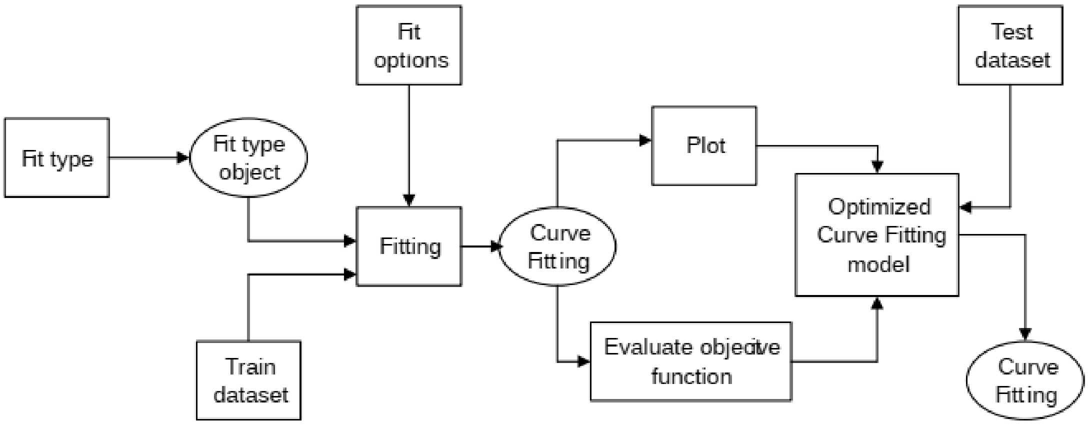


Figure 1 Typical curve fitting approach for prediction

The curve fitting process begins with choosing a fit type. When a fit type is chosen a fit type, object is created. The fit type object together with the train dataset and fit options create a new object Fitting which is passed on to the curve fitting process. The fit options include among others the fitting method, fitting algorithm, convergence criteria and starting lower and upper bounds for coefficients to be determined. During the curve fitting process, the objective function is evaluated and the plot is a visual representation of the process. Since we are using non-linear curves for fit our data, this is an iterative process which results into an optimized curve. Test data is used on the optimized curve fitting model to evaluate its performance. The optimized model is finally used to predict the future. In this study, we developed three curve fitting models using Gauss, Exponential and Fourier functions. The models are shown in the "Equations 3, 4 and 5" respectively.

$$f(x) = a_1 exp^{-(\frac{x-b_1}{c_1})^2} \tag{3}$$

$$f(x) = aexp^{bx} \tag{4}$$

$$f(x) = a_0 + \ a_1 coswx + b_1 sinwx \tag{5}$$

By exploring various parametric and non-parametric fits, the objective is to find appropriate coefficients for the best fit to the data. In this optimization problem, the sum of squared errors (SSE) between the actual value and the predicted value is minimized. Our objective function is thus "as shown in the Equation 6".

$$\min SSE = \ \sum_{i=1}^{n}(Actual\ Load - Predicted\ Load)^2 \tag{6}$$

where n is the number of data points. The goal is to identify the coefficients such that the curve fits the data well. We use the least squares algorithm to solve this optimization problem.

3.2.1. The Least Squares Algorithm

In the least squares fitting process, the summed square of residuals is minimized. The 'best' curve is the one that has minimum error i.e., summed squared error, between data points and the curve. From elementary calculus, the slope of a function is represented by its derivative and the minimum of a function occurs at a point where the slope of the function is zero. Therefore, the least squares algorithm determines the coefficients of the curve fitting function by differentiating the summed squares of residuals with respect to each coefficient, and the result is set equal to zero. In the next paragraphs we describe how the least squares algorithm can be used to determine the coefficients of curve $y = f(x; \beta)$ given the data points (x_i, y_i). Here x is the independent variable whose values are presumed precisely known, and the βj are the n coefficients to be determined.

Let $(x_i, y_i), i = 1,2, \dots, m,$ be a set of data points, and let $y = f(x; \beta)$ a curve to be fitted to this data. At the point $x = x_i$ the experimental value of y is y_i , thus the value that corresponds to the curve to be fitted is $f(x_i; \beta)$: If e_i is the residual of approximated value at $x = x_i$ then $e_i = y_i - f(x_i; \beta)$. The summed square of the residuals S can be written as

$$S = \sum_{I=1}^{m} (y_i - f(x_i, \beta))^2 = \sum_{i=1}^{m} {e_i}^2 \tag{7}$$

Substituting values of y_i and $f(x_i)$ at each x_i in "Equation 7" above, we get S as a function of the coefficients β to be determined. From the theory of calculus, we can determine the minimum of S by taking the partial derivative of S with respect to β_j and equating them to zero, i.e.

$$\frac{\partial S}{\partial \beta_j} = 0 \tag{8}$$

The result is a system of equations having n unknowns. The solution to these equations are the coefficients of the curve/polynomial fitting to the data.

3.2.2. Procedure of the Proposed Model

The method that is proposed is used to find an accurate long-term electrical energy forecasting model for Turkey. The actual annual electrical energy consumption data from 1970 to 2015 was taken from the Turkish Electricity Transmission Company (TETC). Data for the years 1970 to 2005 was used as training data and data the years 2006 to 2015 was used as testing data. Forecasting long term electrical energy consumption is affected by many factors which make it exhibit many non-linearities. Fitting a linear model becomes difficult because the coefficients cannot be estimated using simple matrix techniques. Therefore, the procedure of the least squares' algorithm used for the Gaussian, Fourier, and Exponential models developed in this study follows an iterative process as outlined below.

1. For each coefficient, make and an initial estimate. For our forecasting models, we use random values on the interval [-1,1].
2. For the current set of coefficients, produce the fitted curve.
3. By using a fitting algorithm, we adjust the coefficients to find out if the fit improves. For our study we chose the Levenberg-Marquardt algorithm of the curve fitting toolbox in MATLAB.
4. By returning to step (2), we iterate the process until we reach the specified convergence criteria for our fit.

Various fits are evaluated numerically or graphically to find the best fit. The commonly used performance metrics are based on residuals between the actual and forecasted values. Common metrics include R-Squared, Root Mean Square Error (RMSE), Mean Average Percentage Value (MAPE), Mean Square Error (MSE) and Mean Average Error (MAE) [24]. For this study we evaluate the various fits to the models using R-Squared and RSME. R-squared is also known as the coefficient of determination. R-squared is calculated as shown in the "Equation 9".

$$R^2 = 1 - \frac{SS_E}{SS_X} \tag{9}$$

Where $SS_E$and SS_X are the sum of squares of residuals and errors and given by "Equations 10 and 11", respectively.

$$SS_E = \sum_{i=1}^{n}(x_i - \widehat{x_i})^2 \tag{10}$$

$$SS_x = \sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{11}$$

Where x_i is the actual value, $\widehat{x_i}$ is the forecasted value, and $\bar{x}$ is the mean of the actual values. We use extrapolation to obtain the forecasts.

RMSE is calculated using the formular in "Equation 12" below

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} |x_i - \hat{x}_i|^2}{n}} \tag{12}$$

4. Results

In this study, Gaussian, Fourier and Exponential curve fitting models for long term electricity forecasting were developed. Performance metrics and confidence intervals on fitted coefficients were used to determine how good our models are able to forecast. The performance metrics helped to determine the ability of the curve to fit the data. Descriptively, a high performance metric represents suitability of a certain model is suitable. On the predictive side, a high performance metric implies that the model performs well. The performance metrics (using train data) for the models are shown in the "Table 1".

Table 1 Performance metrics for the forecast models

Model	Performance metric			
	SSE	R-Square	Adjusted R-Square	RMSE
Gauss	617.5	0.9978	0.9977	3.79
Fourier	668.8	0.9976	0.9975	3.99
Exponential	3602	0.9872	0.9869	9.047

Because long term energy forecasting is characterized with lots of uncertainties which affect the accuracy of forecasts, we tried to capture these uncertainties by taking 95% confidence bounds when finding coefficients of curve fitting models. The obtained coefficients are shown in "Table 2".

Table 2 Coefficients of curve fitting models

Model	Coefficients (with 95% confidence bounds)
Guass	a_1 = 433.4 (361.3, 505.4)
	b_1 = 2039 (2034, 2045)
	c_1 = 34.85 (32.16, 37.54)
Fourier	a_0 = 1.1e+008 (-2.8e+013, 2.8e+013)
	a_1 = -1.0e+008 (-2.8e+013, 2.8e+013)
	b_1 = 1.0e+007 (-1.4e+012, 1.4e+012)
	w = -5.1e-005 (-6.7, 6.7)
Exponential	a = 2.6e-052 (-1.2e-051, 1.7e-051)
	b = 0.062 (0.059, 0.065)

Using the optimized coefficients shown in "Table 2", "Equations 3, 4 and 5" become:

$$f(x) = 433.4\ exp^{-(\frac{x-2039}{34.85})^2} \tag{13}$$

$$f(x) = 2.6 * 10^{-52}\ exp^{0.062x} \tag{14}$$

$$f(x) = 1.0 * 10^8 - 1.0 * 10^8 \cos(-5.1 * 10^{-5})\, x + 1.0 * 10^7 sin(-5.1 * 10^{-5}) x \tag{15}$$

The graphs in “Figures 2, 3 and 4” show the Gaussian, Fourier and Exponential curve fits to the data and extrapolations to the 2025, with the consumption in the year 2023 highlighted. We highlight the consumption for the year 2023, because it is the year to which all targets of NDP11 focus on. It is the year the Turkish Republic is making 100 years since its establishment.

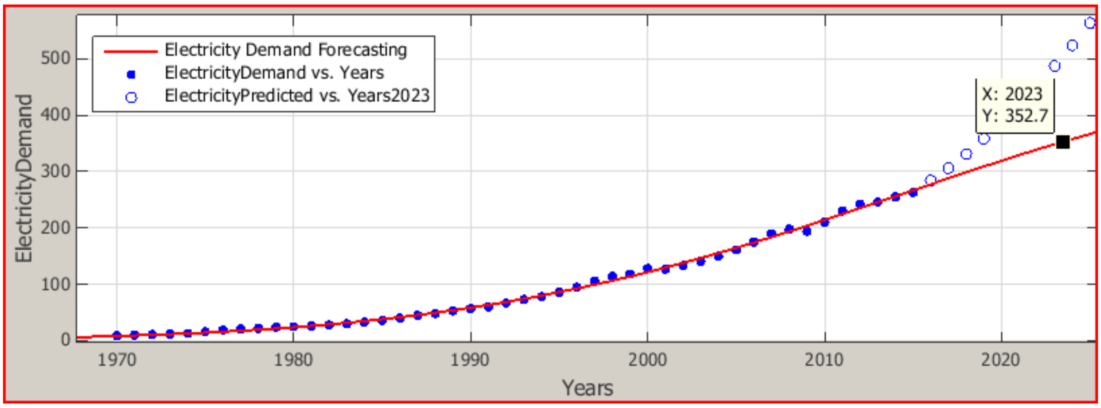


Figure 2 The prediction of electricity demand (TWh) in 2023 using Gaussian Curve Fitting

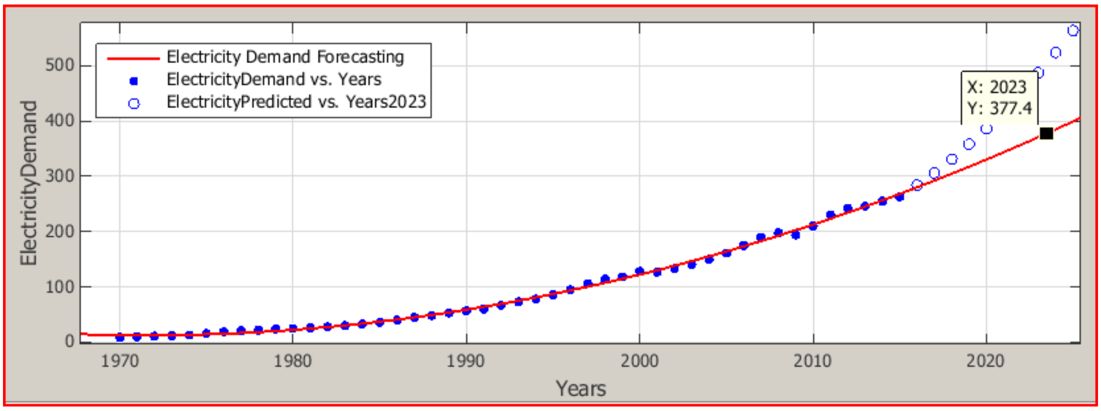


Figure 3 The prediction of electricity demand (TWh) in 2023 Fourier Curve Fitting

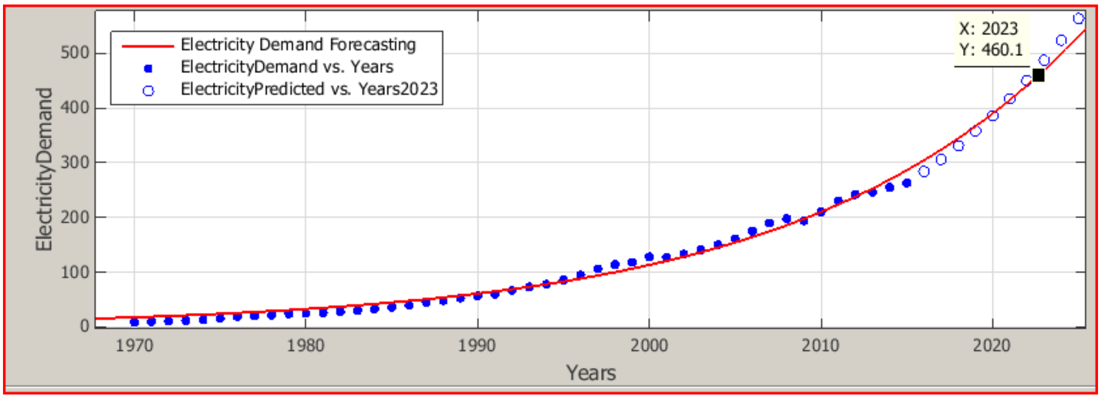


Figure 4 The prediction of electricity demand (TWh) in 2023 Exponential Curve Fitting

The results of the long-term electricity consumption for Turkey are shown in the “Table 3”.

In “Table 4”, we compare our results with those of previous studies. Looking at the comparisons in the “Table 4”, the numerical values of the percentage errors are small for the previous studies but they are negative, meaning that the methods forecast less as compared to the actual. Our study gives bigger forecast errors in comparison to the actual, but they are positive. For planning purposes, it is better to have excess than deficit. This is what makes our study better than the previous studies.

Table 3 Results of forecast models and the actual electricity consumption (TWh)

Years	Actual	GAUSSIAN	FOURIER	EXP	MENR
2015	217.3	277.7	274.9	267.9	267.9
2016	231.2	291.3	286.8	295.3	285.7
2017	249.0	301.1	298.9	319.2	305.7
2018	258.2	310.5	311.4	342.3	327.1
2019	304.3	318.2	326.6	360.5	345.0
2020		323.1	342.2	383.7	374.5
2021		330.9	352.5	407.3	400.7
2022		343.6	360.4	427.3	428.7
2023		**352.7**	**377.4**	**460.1**	**458.6**
2024		360.8	397.9	507.3	490.9
2025		374.3	409.7	520.1	525.2

Table 4 Comparison of our results with previous studies

Year		2015	2016	2017	2018	2019	2015	2016	2017	2018	2019
		Consumption (TWh)					% Forecast error				
Actual		217.3	231.2	249.0	258.2	304.3					
MENR		267.9	285.7	305.7	327.1	345.0	23.3	23.6	22.8	26.7	13.4
Kavaklioglu et al, [25]		212.2	217.7	223.2	228.7	234.3	-2.4	-5.9	-10.4	-11.4	-23.0
Kavaklioglu, [26]		184.1	193.2	199.6	207.6	215.9	-15.3	-16.4	-19.8	-19.6	-29.0
Toksari, [27]		201.5	204.2	206.9	209.5	212.1	-7.3	-11.7	-16.9	-18.9	-30.3
Kiran et al, [28]		249.5	261.2	273.8	287.4	302.1	14.8	13.0	10.0	11.3	-0.7
Our Study	**Guassian**	277.7	291.3	301.1	310.5	318.2	27.8	26.0	20.9	20.3	4.6
	Fourier	274.9	286.8	298.9	311.4	326.6	26.5	24.0	20.0	20.6	7.3
	Exponential	267.9	295.3	319.2	342.3	360.5	23.3	27.7	28.2	32.6	18.5

5. Conclusion

In this study, we have developed models for electrical energy demand forecasting based on a novel curve fitting approach. The study focused on designing a simple yet compact, fast and accurate long term electrical energy forecasting models that can be used for policy formulation and planning especially in the energy sector. Our approach was guided by the principle of finding the minimum error between the observed data and the predicted data. We developed three forecasting models, i.e., Gaussian, Fourier and Exponential curve fitting models. The forecasting models presented in this paper take the year as the independent variable and electrical energy consumption as the dependent variable. The simplicity of the curve fitting models and using only one independent variable results into very low times for both training and forecasting. We used the coefficient of determination (R-Squared) as a measure of forecast accuracy performance. The accuracy of the proposed models is excellent. Accordingly, the Gaussian model gave better results (R-squared = 0.9978) followed by the Fourier (R-squared = 0.9976) and Exponential (R-squared = 0.9872) models respectively. We have been able to capture uncertainties that exist in every forecasting model by taking 95% confidence values for the coefficients of the Gaussian, Fourier and Exponential models to get a lower and upper bound forecast. Accordingly, our forecasts from the Gaussian, Fourier and Exponential models for the target year 2023 are 352.7 TWh, 377.4TWh and 460.1 TWh respectively. Except for the Exponential model, the forecasts for the Gaussian and Fourier models are close to and in agreement with the estimated 375.8 TWh energy demand in NDP11.

Acknowledgment

The authors thank Mr. Mehmet Alkan for his valuable help in simulation.

References

[1] C. Erdin, and G. Ozkaya, “Turkey’s 2023 Energy Strategies and Investment Opportunities for Renewable Energy Sources,” Site Selection Based on ELECTRE. Sustainability, vol. 11, pp. 21-36, 2019.

[2] Turkey Electricity Transmission Company, Electricity Generation - Transmission Statistics of Turkey, 2018.

[3] Turkey Electricity Distribution Company, Electricity Distribution and Consumption Statistics of Turkey, 2018.

[4] Presidency of the Republic of Turkey, Eleventh Development Plan (2019-2023), July 2019

[5] Y. Yunus and J. Mo, “Forecasting of Turkey’s Electricity Consumption Using Artificial Neural Network,” World Automation Congress 2014 ISI Press.

[6] A. Gülsüm, “Forecasting Regional Electricity Demand for Turkey,” International Journal of Energy Economics and Policy, vol. 7, no. 4, pp. 275-282, 2017.

[7] B Şule, T. Ayça, “Modelling and Forecasting Turkey’s Electricity Consumption by using Artificial Neural Network,” American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS), vol. 25, no. 1, pp. 192-208, 2016.

[8] A. Gokhan, “Modeling of energy consumption based on economic and demographic factors: The case of Turkey with projections,” Renewable and Sustainable Energy Reviews, vol. 35, pp. 382-389, 2014.

[9] D.C. Sansom, T. Downs, T. K. Saha, “Evaluation of support vector machine-based forecasting tool in electricity price forecasting for Australian national electricity market participants,” Journal of Electrical and Electronics Engineering Australia, vol. 22, no. 3, pp. 227-234, 2003.

[10] C. Kuster, Y. Rezgui, M. Mourshed, “Electrical load forecasting models: A critical systematic review,” Sustainable Cities and Society, vol. 35, pp. 257-270, 2017.

[11] A.T. Barran, A. Alonso, J.R. Dorronsoro, “Regression tree ensembles for wind energy and solar radiation prediction,” Neurocomputing, vol. 326, pp. 151-160, 2019.

[12] O. F. Ertugrul, “Forecasting electricity load by a novel recurrent extreme learning machines approach,” Electrical Power and Energy Systems, vol. 78, pp. 429-435, 2016.

[13] J. Zhang, Y.M. Wei, D. Li, Z. Tan, J. Zhou, “Short term electricity load forecasting using a hybrid model,” Energy, vol. 158, pp. 774-781, 2018.

[14] A. Khosravi, R. N. N. Koury, L. Machado, J.J.G. Pabon, “Prediction of wind speed and wind direction using artificial neural network, support vector regression and adaptive neuro-fuzzy inference system,” Sustainable Energy Technologies and Assessments, vol. 25, pp. 146-160, 2018.

[15] A. Unler, “Improvement of energy demand forecasts using swarm intelligence: The case of Turkey with projections to 2025,” Energy Policy, vol. 36, pp. 1937-1944, 2008.

[16] M. Kankal, E. Uzlu, “Neural network approach with teaching learning-based optimization for modeling and forecasting long-term electric energy demand in Turkey,” Neural Comput and Applic, vol. 28, no.1, pp. 737-747, 2017.

[17] İ. Topcu, F. Ülengin, Ö. Kabak, M.Işık, B. Ünver, Ş.Ö. Ekici, “The evaluation of electricity generation resources: The case of Turkey,” Energy, vol. 167, pp. 417-427, 2019.

[18] F. Kaytez, “A hybrid approach based on autoregressive integrated moving average and least-square support vector machine for long-term forecasting of net electricity consumption”, Energy, vol. 197, pp. 187-200, 2020.

[19] H.H. Çevik, H. Harmancı, M. Çunkaş, “Forecasting Hourly Electricity Demand Using a Hybrid Method,” 2017 International Conference on Consumer Electronics and Devices.

[20] C. Hamzaçebi, H.A. Es, R. Çakmak, “Forecasting of Turkey’s monthly electricity demand by seasonal artificial neural network,” Neural Comput and Applic, vol. 31, pp. 2217-2231, 2019.

[21] G. Çeribaşı G, M. Çalışkan, “Short and long term prediction of energy to be produced in hydroelectric energy plants of Sakarya Basin in Turkey,” Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, pp. 1-16, 2019.

[22] M.A. Farahat, M. Talaat, “A New Approach for Short-Term Load Forecasting Using Curve Fitting Prediction Optimized by Genetic Algorithms” Mathematics and Engineering,” pp. 19-21, 2010.

[23] E. Ismail, B. Rachid, A. Abdelah, A. Othman, M.G. Josep, “Energy Production: A Comparison of Forecasting Methods using the Polynomial Curve Fitting and Linear Regression” International Renewable and Sustainable Energy Conference (IRSEC), Tangier, pp. 1-5, 2017.

[24] R. J. Hyndman and A.B. Koehler, “Another look at measures of forecast accuracy” International Journal of Forecasting vol. 22, pp. 679– 688, 2006.

[25] K. Kavaklioglu, H. Ceylan, H.O. Kemal, O.E. Canyurt, “Modeling and prediction of Turkey’s electricity consumption using Artificial Neural Networks,” Energy Conversion and Management, vol. 50, pp. 2719-2727, 2009.

[26] K. Kavaklioglu, “Modeling and prediction of Turkey’s electricity consumption using Support Vector Regression,” Applied Energy vol. 88, pp. 368-375, 2011.

[27] D. Toksarı, “Estimating the net electricity energy generation and demand using the ant colony optimization approach: Case of Turkey,” Energy Policy, vol. 37, pp. 1181-1187, 2009.

[28] M.S. Kıran, E. Ozceylan, M. Gündüz, T. Paksoy, “Swarm intelligence approaches to estimate electricity energy demand in Turkey,” Knowledge-Based Systems, vol. 36, pp. 93-103, 2012.