Using Hierarchical Performance Modeling to Determine Bottleneck in Pattern Recognition in a Radar System

Ahmed Alsheikhy and Muhannad Almutiry
aalsheikhy@nbu.edu.sa *muhannad.almutiry@nbu.edu.sa* *ahmed.alsheikhy@gmail.com*
Electrical Engineering Department, College of Engineering, Northern Border University, Arar, Saudi Arabia

Summary
The radar tomographic imaging is based on the Radar Cross-Section "RCS" of the materials of a shape under examination and investigation. The RCS varies as the conductivity and permittivity of a target, where the target has a different material profile than other background objects in a scene. In this research paper, we use Hierarchical Performance Modeling "HPM" and a framework developed earlier to determine/spot bottleneck(s) for pattern recognition of materials using a combination of the Single Layer Perceptron (SLP) technique and tomographic images in radar systems. HPM provides mathematical equations which create Objective Functions "OFs" to find an average performance metric such as throughput or response time. Herein, response time is used as the performance metric and during the estimation of it, bottlenecks are found with the help of OFs. The obtained results indicate that processing images consumes around 90% of the execution time.

Key words:
hierarchical performance modeling, bottlenecks, tomographic images, radar systems, single layer perceptron.

1. Introduction

Radar is an electromagnetic sensor device that is used to track, recognize, detect and locate different targets [1,2]. It detects and determines the velocity, angle and range of an object using electromagnetic waves or spectrums [1,3,4,5,6,10]. Every radar contains several components to operate perfectly and those components are a transmitter, an antenna, a receiver and a processor unit [1,4,5,6], in general, the antennal works for transmitting and the receiving. Radars are used for locating aircraft, ships and giving information about weather. Nowadays, radars produce radio frequencies to obtain information about objects. Tomography is the obtained images from radars. Tomography images are generated in radars using multistatic methods [5,6]. These methods provide useful data and information about shapes and edges of targets being tracked and considered.

Numerous sensing applications such as Ground-Penetrating Radar (GPR), Human Activities Recognition (HAR) and respiratory disorder use the radar tomography technique [1,6,7]. However, this method experience difficulties from weak scatters near strong scatters when using a masking approach [6,7,8,9]. However, several approaches were developed and implemented to overcome that issue such as the CLEAN algorithm [6]. It cleans the dominant scatters from a received wave to lower the effect of Sidelobe in the radar tomography. Using radiation waves presents numerous scattering fields levels that are relevant to the shape of targets [6]. In tomographic radars, several positions of the targets can be seen and found because of using the spatial distribution of transmitters and receivers [6,7,8,9].

The identification of the targets happens through the generating of images with a high resolution which people can observe them or by the creation of a target representation which can be recognized by machines [1]. Tomographic approaches incorporate polarization degrees of freedom, angular and distance to represent a close and actual view of a target's shape. Generating the actual physical aperture can be too expensive [1]. Using Neural Networks (NN) in perceiving the target recognition in the time domain has got the attention of researchers. High complexity and noise have occurred during the process of the target recognition in the time domain [1]. Recently, researchers have proposed the NN in natural frequency based as an independent angle [1,6,8,9,10]. This method yields less noise immunity.

In particular, several algorithms to schedule multiple periodic tasks exist which are A) Rate Monotonic (RM), B) Deadline Locally Linear Embedded (LLE) algorithms are preferred for data dimension reduction due to their robustness for determining and addressing the difficulties of nonlinear dimensionality reduction [1]. These issues/difficulties emphasize the hardness of data mapping onto a low dimension space when mapping from a low-dimensional manifold in a high-dimensional data space [1,2]. Recently, Multidimensional Scaling (MS) and Principal Component Analysis (PCA) approaches have been developed to have linear dimensional reduction. However, it has been noticed that such methods suffer when a nonlinear manifold occurs [1]. Thus, using unsupervised LL is practically preferred since it addresses the difficulties of high-dimensionality reduction. This method is capable of finding low-dimensional data from high one. In addition,

Manuscript received March 5, 2022
Manuscript revised March 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.3.38

several advantages can be found from using LLE methods such as the ability to consume fresh data and the speed of implementation [1].

Authors in [1] proposed a novel method for pattern recognition by using Multi-Layer Perceptron (MLP) on radar tomographic images in the frequency domain. Using LLE on MLP with appropriate signatures of the targets yields a promising high-resolution target identification. Nevertheless, the same result can be obtained even when the information is incomplete [1]. Combining both methods lead to remove the need of recognizing the expensive apertures of remote targets. It shows that identifying targets through a single operation frequency is possible when nonlinear dimensional dynamic systems are used [1].

Our contribution in this paper is to determine and spot the bottlenecks that occur when using LLE and MLP together for pattern recognition of handwritten digits for numbers from 0 to 9 as a frequency response using tomographic radars. Adjusting the proposed approach in [13] is performed to construct performance objective functions/equations that are used later to determine the bottlenecks. Those objective functions are the result of merging a functional approach with a mathematical one. For the functional approach, a Hierarchical Generic Finite State Machine was developed. On the other hand, the Hierarchical Performance Modeling approach developed in [16,17] is used as the mathematical one. Readers can refer to [13,14,16,17] for more information. The proposed approach is capable to estimate and predict several performance metrics such as execution, power consumption and throughput. However, finding the bottlenecks is considered in this paper as stated earlier. Our investigations confirm that the bottlenecks occur when converting data from the high-dimensional level onto the low one as almost 80% of the needed time is consumed in that phase.

In the reminder of this paper, section 2 presents the related work on performance evaluation of pattern recognition using radars. Follows by explaining the approach in detail to determine the bottlenecks in section 3. In section 4, numerical evaluations are presented. Lastly, the conclusion of the paper is in section 5.

2. Related Work

Jastrzebska A. in [3] proposed a method to classify time series using a visual pattern recognition. This approach converted the scalar time series values to the amplitude of two-dimensional space and then incremented those values subsequently. Once the time series classification issue was transferred to the visual pattern recognition, then the author demonstrated the new approach through several experiments available time series. Comparing the obtained results with numerous recent and unique methods dedicated for time series classification was performed. These tests proved that the proposed approach was robust and they showed that its steadiness. In addition, accepted and satisfying accuracy was achieved. More information is found in [1]. In this paper, constructing the objective functions is done first, then spotting the bottlenecks through those functions can be easily found. Any part from the objective functions with the highest value implies that the bottlenecks occur in it.

B. George et al. in [4] evaluated numerous performance parameters which were minimum detection signal, minimum signal-to-noise ratio, maximum detection range equation and lastly system loss. Furthermore, additional parameters were used and investigated too. Then, MATLAB as a simulation tool was used to compute those parameters which concluded that the detection signal of targets can be issued when the signal-to-noise ratio is low. In this paper, several performance parameters can be calculated such as the execution time, which is also known as delay/latency, energy dissipation and throughput. Herein, only latency is considered to determine the bottlenecks. Readers can find more information in [4].

The authors in [5] used the signal-to-noise ratio (SNR) as a metric to evaluate the performance of a high-resolution radar system. Two cases were considered in that evaluation which were four-channel architecture and eight-channel one. Authors set the peak power of the pulsed source to vary from 10dbm to 20dbm and the distance between the antenna and sensing object was 2.4m. Authors found that when the distance increased to 13.5m, the SNR ranged between 9db and 15db for four-channel architecture and between 11db to 16db in another one. Furthermore, SNR could be reduced when optical pulsed sources with more flattened spectrum are used. However, the authors did not pay attention to the delay as a parameter in their research. In this paper, the execution time is evaluated numerically with the help of the obtained objective equations from the developed framework in [13].

D. Nagarajan and G. Kavitha in [7] recognized the pattern of IRIS flower by Neural Network (NN) time series to identify the petal and sepal size. NN was used to predict the desired behavioral pattern in IRIS species. In that paper, time and the gradient parameters were examined,10 epochs were tested and the less obtained time was in the 4th epochs and it was around 0.122s. In this paper, the delay is computed to determine the bottlenecks so designers can improve the design to reduce the occurred delay.

3. Research Methodology

The motivation of this research is to spot any bottlenecks that may occur in tomographic radars so concerned teams can modify their designs or functions inside them to achieve better execution time which implies that the bottlenecks were reduced significantly. In addition, minimizing the potential cost of a radar diversity imaging system is also possible if concerned teams are able to identify the place or places of bottlenecks which can be achieved by applying the implemented approach in [13]. In [1], the proposed method is capable to recognize the patterns even the information is incomplete.

Generating several signatures or attributes of any target such as its velocity, location and range is based on several factors which are not handicapping ones [1]. Earlier experiments on the NN proved that it could take sparse information whether the scattering objects were complex or just simple and produced the high-resolution recognition [1]. Tomographic radars can determine the targets from different angles based on the locations of their transmitters and receivers. The amount of needed information can be expanded or elevated due to the diversity of geometric distributions [1,2,3]. In general, tomographic radars are composed of various dipoles transmitters (N) and various dipoles receivers (M) [1]. Fig. 1 depicts an area to be examined with an individual pair of dipoles, transmitter and receiver. In fig. 1, $\boldsymbol{r_n^t}$ denotes the location of transmitter n while $\boldsymbol{r_m^r}$ represents the location of receiver n. In addition, $\hat{\mathbf{a}}_\mathbf{n}^\mathbf{t}$ and $\hat{\mathbf{a}}_\mathbf{m}^\mathbf{r}$ refer to the polarization of both dipoles respectively. Furthermore, having a free space medium and unknown contrast function $\tau_\delta(r')$ were considered as stated in [1] where r' denotes a position vector.

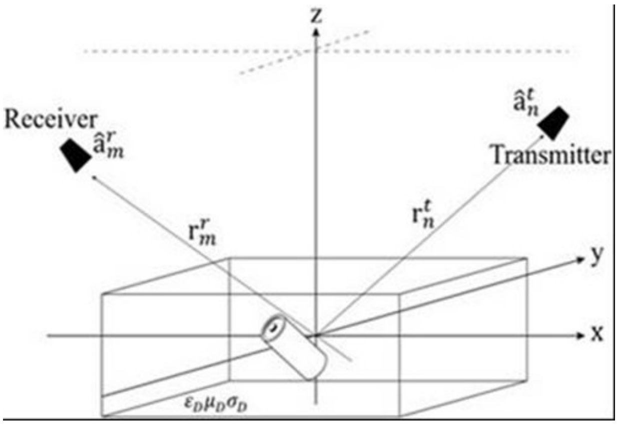

Fig.1 representation of a target as 3D model

The scattered field equation of transmitter n and receiver m $\mathbf{E}^S(\boldsymbol{r_n^t}, \boldsymbol{r_m^r})$ is illustrated as follows in which utilizing Born approximation is applied [1]:

$$\mathbf{E}^S(\boldsymbol{r_n^t}, \boldsymbol{r_m^r}) = k_0^2 \iiint_D [\hat{\mathbf{a}}_\mathbf{m}^\mathbf{r} \cdot \overline{\mathbf{G}}(\boldsymbol{r_m^r}, \mathbf{r}')] \cdot [\overline{\mathbf{G}}(\mathbf{r}', \boldsymbol{r_n^t}) \cdot \hat{\mathbf{a}}_\mathbf{n}^\mathbf{t}] \tau_\delta(\mathbf{r}') d\mathbf{r}' \quad (1)$$

k refers to a wavenumber while $\overline{\mathbf{G}}(\boldsymbol{r_m^r}, \mathbf{r}')$ denotes the Green's function. The above equation can be expressed in terms of a matrix multiplication since the Born approximation provides a linear relationship with the contrast function as follows [1]:

$$\mathbf{E}^S(\boldsymbol{r_n^t}, \boldsymbol{r_m^r}) = \mathbf{L}(\tau_\delta(\mathbf{r}')) \quad (2)$$

The forward model can be developed using the multiplication of an L matrix. The operator L should sustain the inversion computation of the considered scattering field which implies that generating the scattering field from tomographic images is the only needed to perform the low-dimensional method. In the NN approach, preprocessing operation is required to attain data representations such as clustering, classification and visualization. In addition, all gathering data is descriptive and useful too [1]. Using either supervised or unsupervised methods is the only available approach to gain data dimensionality reduction [1]. Nevertheless, the unsupervised technique is used for most practical cases. In addition, it is considered a time consuming and costly method.

3.1 Unsupervised Locally Linear Embedding (LL)

This approach is summarized as follows:

(1) Let X be a scattering field vector with a set of N points in high-dimensional space R^D so

$$X = \{x_1, x_2, \dots x_N\} \quad (3)$$

(2) Let Y be the frequency response sample of the handwritten digits with N samples in the space R^D so

$$Y = \{y_1, y_2, \dots y_N\} \quad (4)$$

Keep in mind that the frequency response is seen as a nonlinear dimensional of the manifold where d < D, commonly, d <<D and D denotes the received scattering field from numerous angles. This D is represented as a set of bandwidths.

(3) Convert the multidimensional data D into one coordinate system. In [1], D is set to 841. A refers to the number of classes and B represents the number of patterns. Each pattern is seen as a $D \times 1$ vector, so

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_D \end{bmatrix} \quad (5)$$

(4) Finding the K nearest neighbors is calculated using the Euclidean Distance equation for every x_i where $x_i \epsilon X$. So

$$\text{Distance} = \| X_i - X_j \|^2 \quad (6)$$

(5) The resultant matrix of $k \times N$ is the Proximity matrix where the first row refers to the indices

of all neighbors of the first pattern while the second row refers to the second pattern and so on as illustrated in fig. 2.

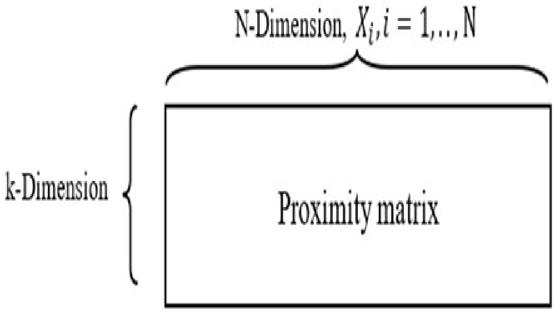

Fig. 2 Proximity matrix

(6) The weight W_{ij} is calculated as follows:

$$\widetilde{X_i} = \sum_{j=1}^{N} W_{ij}\, X_j \quad (7)$$

Where every point X_i is defined as a combination of all its weighted neighbors. W_{ij} is zero when both points are not neighbors. In addition, it is a square matrix with a dimension of $N \times N$. So W_{ij} can be formed and represented as follows:

$$W_{ij} = \begin{bmatrix} w_{11} & \cdots & w_{1N} \\ \vdots & \ddots & \vdots \\ w_{N1} & \cdots & w_{NN} \end{bmatrix} \quad (8)$$

(7) The error function in terms of weight W is computed as follows:

$$E(W_i) = \left\| X_i - \sum_{j=1}^{N} W_{ij}\, X_j \right\|^2 \quad (9)$$

While the total error is defined as follows:

$$E(W) = \sum_{i=1}^{N} \left\| X_i - \sum_{j=1}^{N} W_{ij}\, X_j \right\|^2 \quad (10)$$

(8) The matrix A is updated using the following equation:

$$A = (I - W)^T (I - W) \quad (11)$$

In the above eq. A, I and W are matrices with a dimension of N * N and I represents the identity matrix.

(9) Lastly, Eigen analysis is performed on the resultant A matrix to find d eigenvectors which are the representation of the lowest non-zero values. d is the number of obtained classes A – 1.

MLP network is used to classify the low dimensional data to proper classes, where the input of MLP is Y_i with d-dimension, and the output is A = 10 classes.

As stated earlier, the developed framework in [13] is used in this research to obtain the objective functions which will help us to spot the occurred bottlenecks. Fig. 3 displays a Hierarchical Generic Finite State Machine (HGFSM)in [15] which is used to integrate with an analytical approach, Hierarchical Performance Model (HPM) to form an integrated model after concerting both methods to a Markovian scheme. Readers refer to [13,15] for more information about the developed framework and its affiliated HPM.

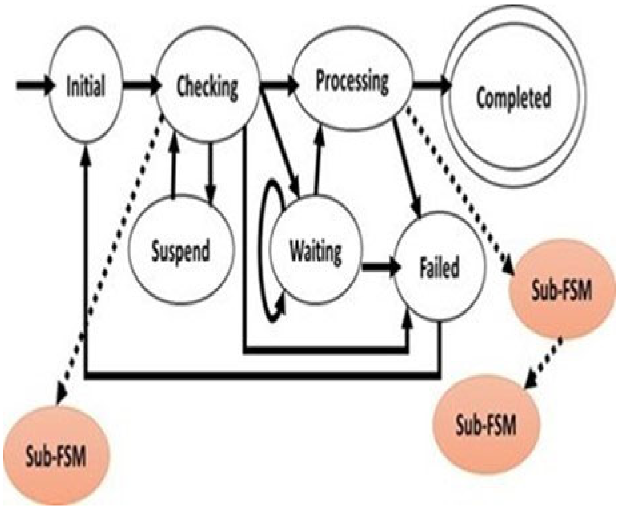

Fig. 3 HGFSM

Fig. 4 illustrates the HPM layers and identifies which operations take place in every layer. In the below figure, execution time evaluation is considered, however, the HPM can be applied on any system and any performance parameter or metric can be computed.

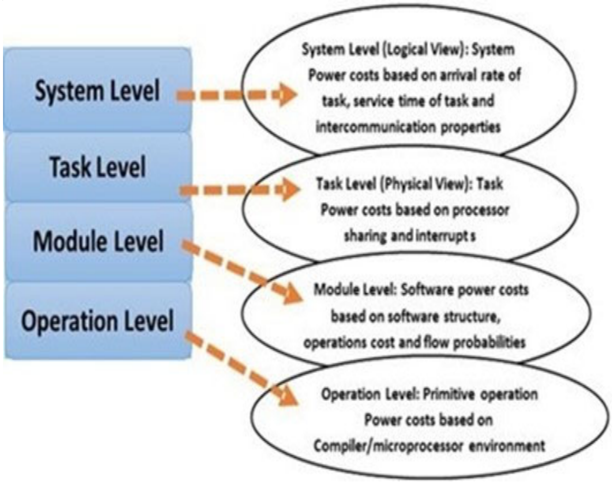
System Level
Task Level
Module Level
Operation Level
System Level (Logical View): System Power costs based on arrival rate of task, service time of task and intercommunication properties
Task Level (Physical View): Task Power costs based on processor sharing and interrupt s
Module Level: Software power costs based on software structure, operations cost and flow probabilities
Operation Level: Primitive operation Power costs based on Compiler/microprocessor environment

Fig. 4 HPM stack layers and operations in every layer

In this part, making adjustments to the developed framework to be applied on tomographic radars is performed. Mapping the proposed approach in [1] to the framework in [13] is achieved by linking every procedure in [1] to its allied phase in HGFSM as follows:

In the Initial state/phase: identifying the neighbors for every input of the considered data to construct K matrix. Then, the proximity matrix is created based on engaging each neighbor for each input. The dimension size of that matrix is K X N, so it is 18 X 1000. Furthermore, finding the weight for every pair of neighboring points. The dimension size for the weight matrix is N X N, so it is 1000 X1000.

In the Checking state/phase: the system evaluates the stipulation (total summation of weights equals 1) which is required to satisfy the circulation, adaptation and scaling operations. More information can be found in [1].

In the Waiting state/phase: radars process an image at a time. Hence, no job/task is sent to this phase so $C_{wait} = 0$.

In the Processing state/phase: the calculation of the low-dimension Y_i is accomplished through the minimization of a function $\Phi(Y)$ as stated in [1]. In addition, two conditions should be met to reach the acceptable minimization level as stated in [1]. Those two conditions are $\sum_{i=1}^{N} y_i = 0$ and $\frac{1}{N} \sum_{i=1}^{N} y_i\, y_i^T = I$. After that, the computation of Matrix A is done where it is a square matrix with dimension $N \times N$. Then using eigenvalue is applied on the obtained matrix A where eigenvector is of size d = 9. The resultant from the previous procedure is the low-dimensional set of data Y_i where its dimension is $d \times N$, 9×1000.

In the proposed framework, probability is used as it is required in the HPM approach and it is computed as follows:

$$P_{ij} = S_{ij}/Z_i \qquad (12)$$

Where P refers to the possibility of going from a source to a destination in the HGFSM, S represents the total number of tasks that move to their destination phase in the HGFSM while Z represents a total number of tasks in the source state. In addition, both subscripts (i and j) refer to the indices of source and destination in the HGFSM. Readers can refer to [13,15] for more information. In our case, P = 1 as all images are sent to the next phase in the developed framework to process them.

To determine the objective functions to spot the bottlenecks in the consider system, a Computation Structure Method (CSM) is required, much information about it is available in [13,15, 16,17,18]. Hence, the objective function equation is as follows:

$$OBF=(1*C_{initial})+((1+e_4)*(C_{check}+C_{test}))+((e_{11}+1)*C_{decision})+((e_9+e_8)*(C_{wait}+C_{test}))+((e_{11}+1)*(C_{exe}+C_{test})) \qquad (13)$$

In the above eq., C denotes the value of cost associated with all operations being held in every phase in the HGFSM while *e* is called the flow variable and it refers to a value of traveling through a certain path in the CSM. The path starts from a node called the start node and it ends with another node called the end node. In addition, the resultant value from the previous eq. represents the average one.

Table 1 lists all primitive operations and shows the average execution time in μs for every operation. In addition, all results were obtained from conducting several experiments on a simulation tool. In this case, MATLAB was used. However, the execution time varies from a platform to another one based on numerous factors such as CPUs and their technologies.

Table 1: Primitive operations and their times

Primitive operations	Average value
Function call	27
Addition	12
Subtraction	39
Multiplication	65
Division	83

4. Results and Discussion

In this experimental simulation, two classes for images are available which are unsupervised and supervised Handwritten digits from 0 to 9 were considered as the measurement domain as stated earlier. Fig. 5 to fig. 7 show three examples of handwritten digits from the MNIST data set. Every image in the below figures is set to 28 X 28 pixels and each pixel is 0.0037 X 0.0037 m. The frequency is set 8Ghz to scan the handwritten digits. 28 transmitters and 28 receivers are used in the simulation experiments and placed to form circles around the target. Every transmitter is set to be 2m away from the center while every receiver is set to be 0.4m away as depicted in fig. 8. Fig. 9 to fig. 11 illustrate the power density of the handwritten digits, 0, 4 and 5, which clearly shows the diversity of their shapes in the frequency domain. In addition, fig. 12 to fig. 17 display the phase and the eigen function for the same three digits.

Before starting the simulation experiments, the number of data input is set to N = 1000 and the number of classes A is 10 while d is 9 = 10-1 for the low-dimensional sets. Furthermore, the number of neighbors K is set to 18.

For the training phase, every handwritten digit is associated with its 100 scattering field images. Hence, 1000 images are used while every digit is associated with its 10 images in the testing phase. Figure 18 illustrates the percentage of obtained error versus the number of iterations during the simulation tests. The error converges to almost zero (0) as the number of iterations increases.

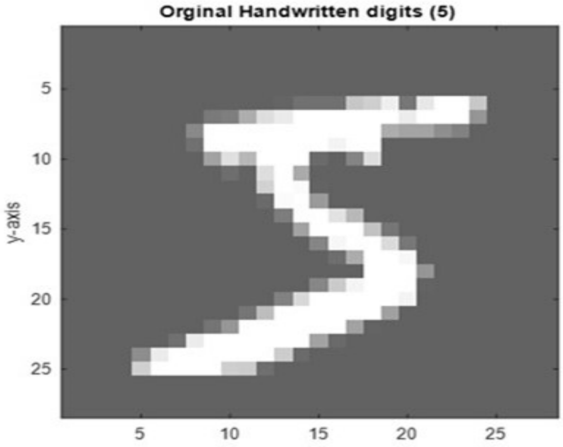

Fig. 5 Handwritten digit 5

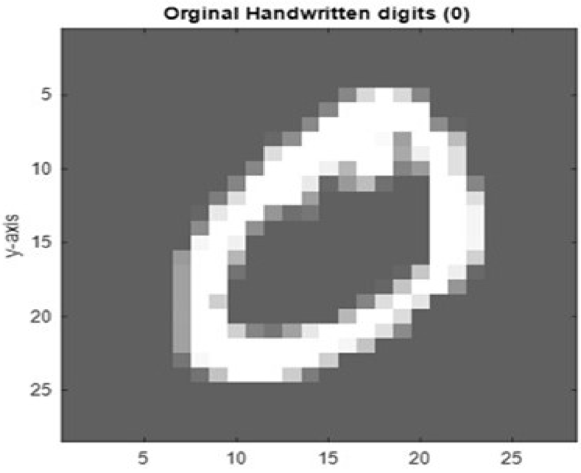

Fig. 6 Handwritten digit 0

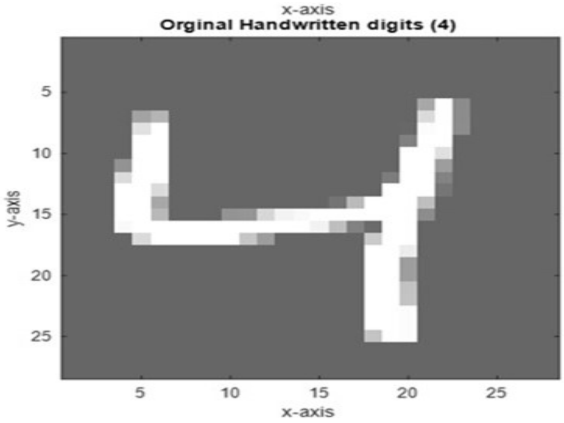

Fig. 7 Handwritten digit 4

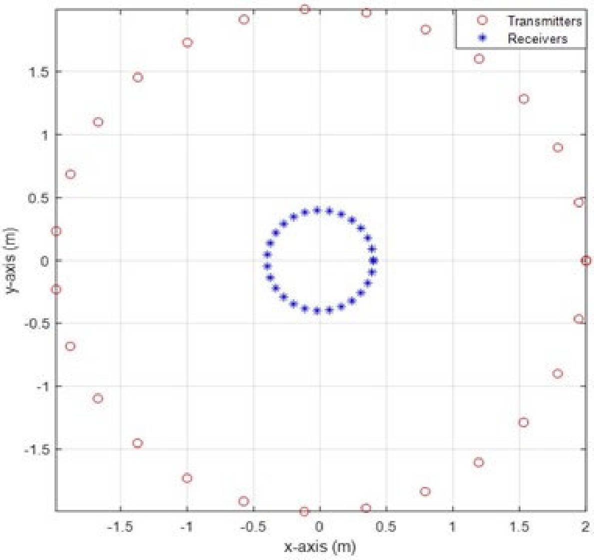

Fig. 8 The placement of the transmitters and receivers

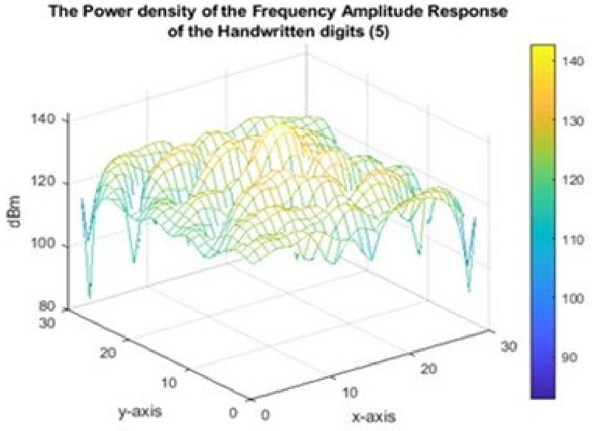

Fig. 9 The power density shape of digit 5

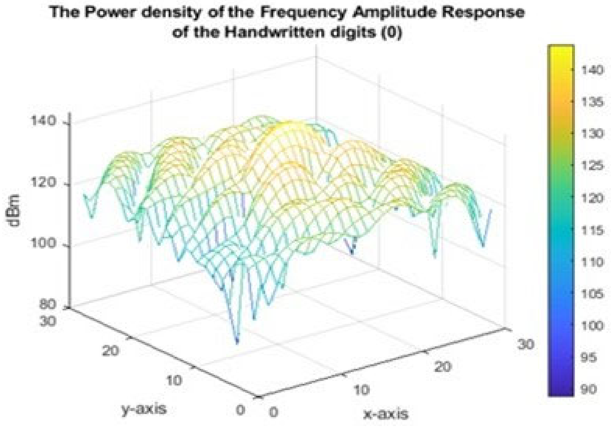

Fig. 10: The power density of digit 0

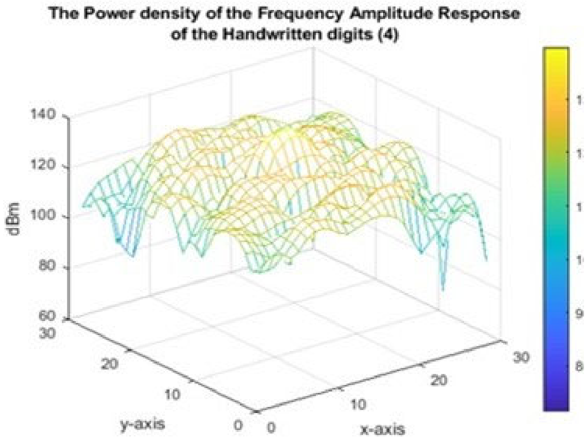

Fig. 11 The power density of digit 4

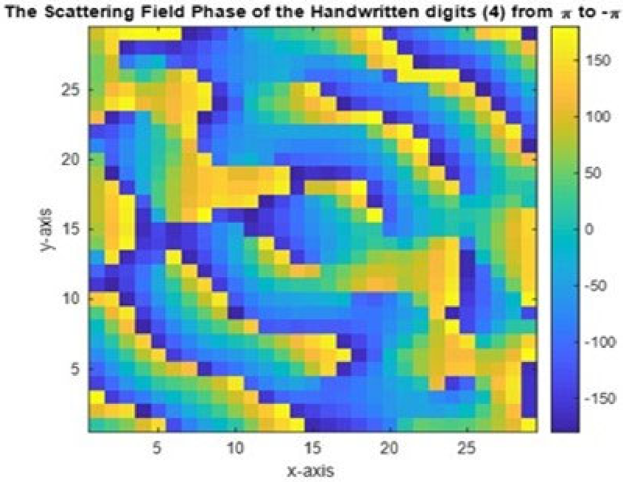

Fig. 14 The phase shape of digit 4

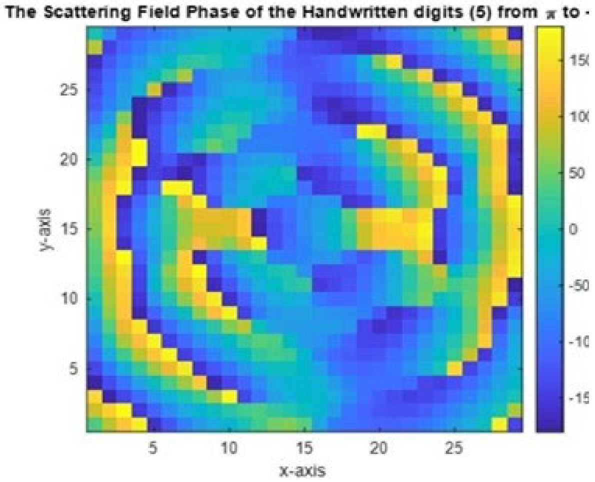

Fig. 12 The phase shape of digit 5

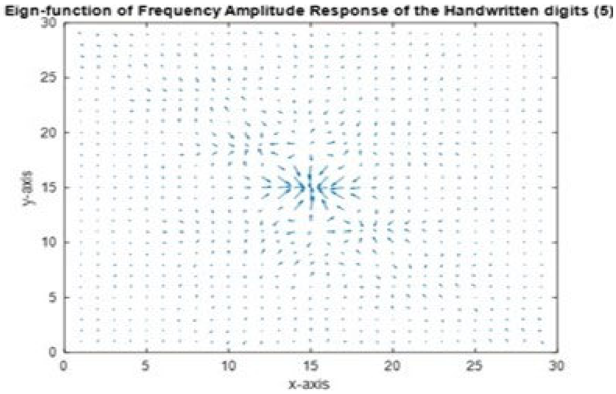

Fig. 15 The eigen function of digit 5

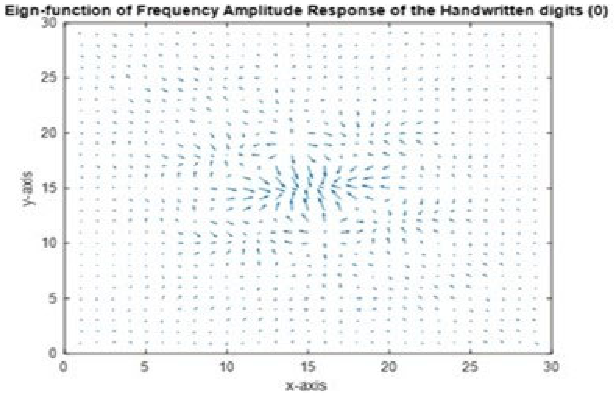

Fig. 16 The eigen function of digit 0

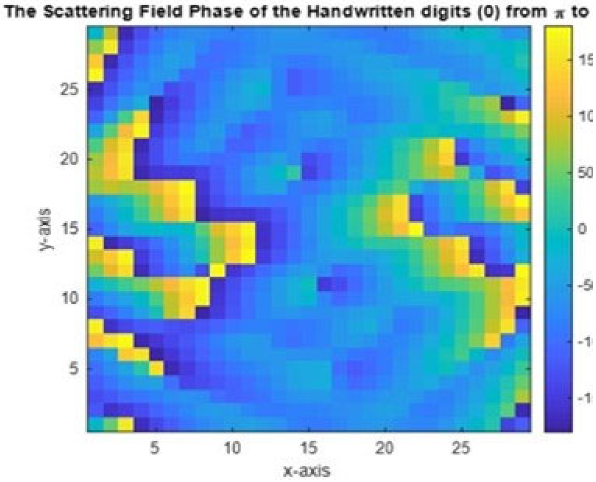

Fig. 13 The phase shape of digit 0

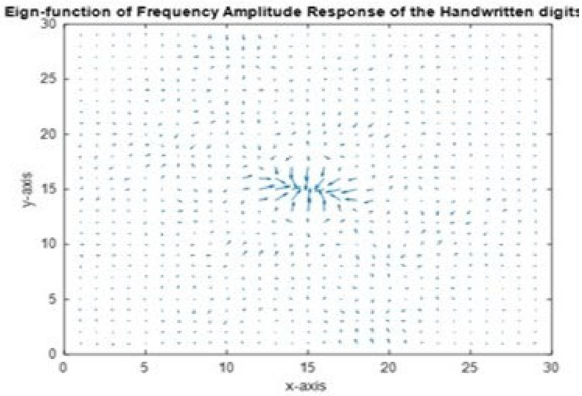

Fig. 17 The eigen function of digit 4

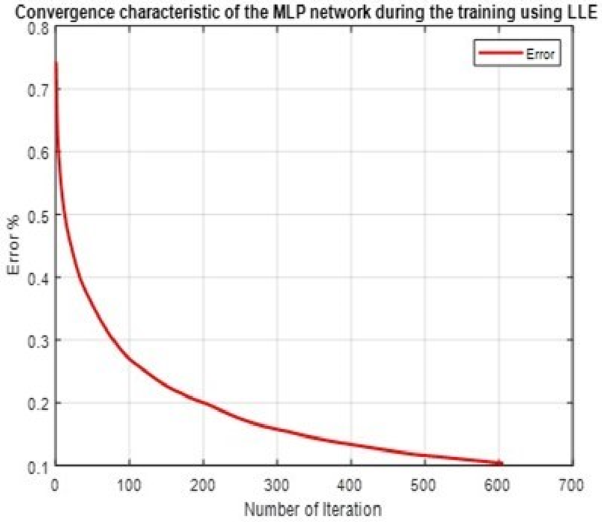

Fig. 18 Convergence characteristic of MLP using LLE

Fig. 19 to fig. 21 display Control Flow Graphs (CFG) of all phases to compute the cost of their operations. These CFGs are part of CSM. More information can be found in [13,15,16,17,18]. The below figs. depict which operations occur there.

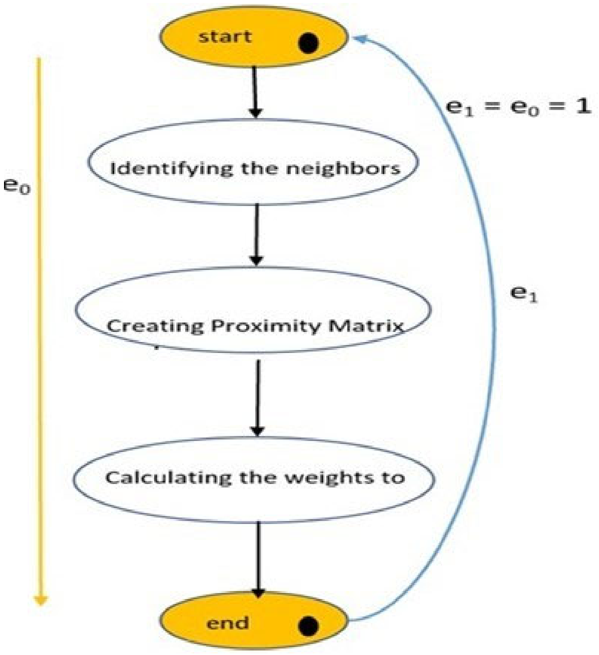
start
Identifying the neighbors
Creating Proximity Matrix
Calculating the weights to
end
e_0
$e_1 = e_0 = 1$
e_1

Fig. 19 CFG for the Initial phase

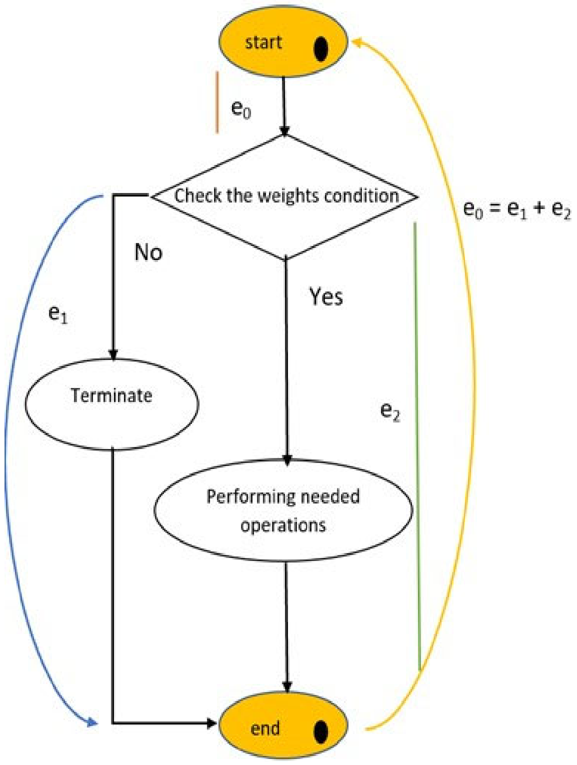

Fig. 20 CFG for the Checking phase

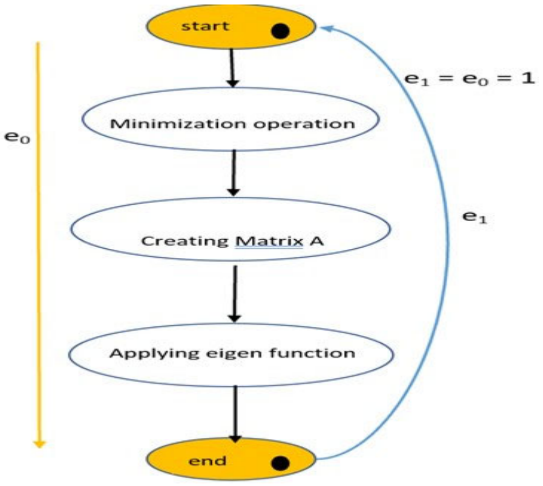

Fig. 21 CFG inside the Processing phase

The following charts display the average execution time that is spent in every phase for all internal operations for one image for unsupervised and supervised LLE. The time is measured in ms.

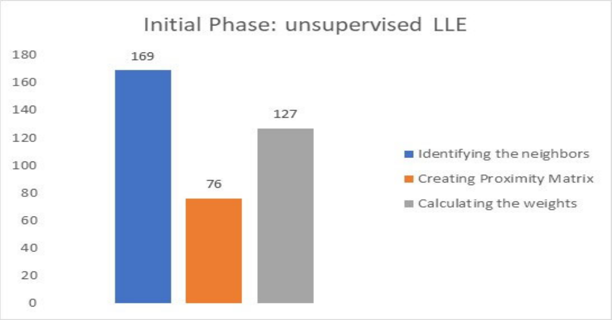

Fig. 22 Average execution time for unsupervised LLE in the Initial state

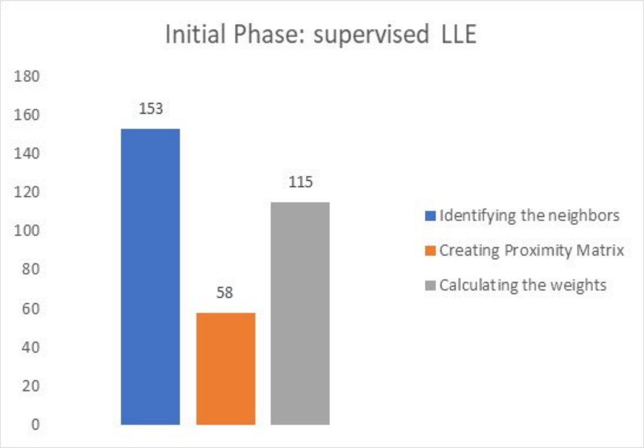

Fig. 23 Average execution time for supervised LLE in the Initial state

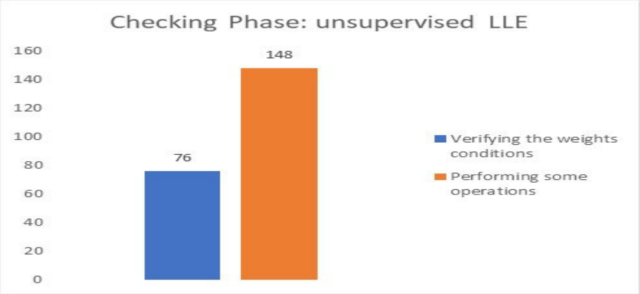

Fig. 24 Average execution time for unsupervised LLE in the Checking state

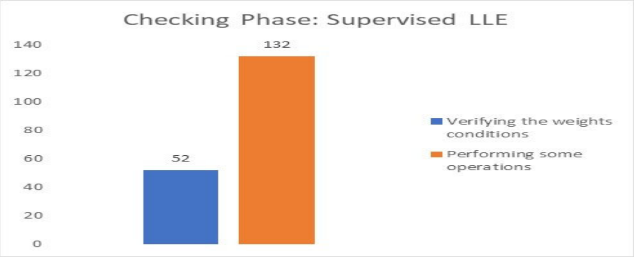

Fig. 25 Average execution time for supervised LLE in the Checking state

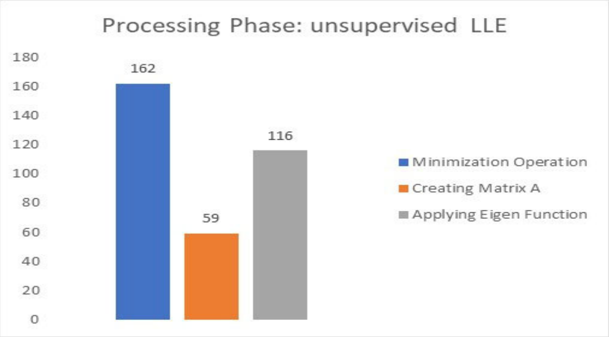

Fig. 26 Average execution for unsupervised LLE time in the Processing state

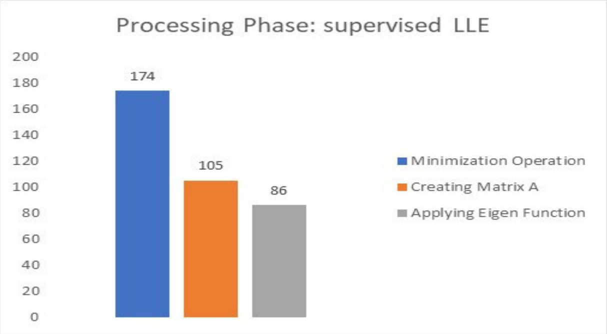

Fig. 27 Average execution for supervised LLE time in the Processing state

From the previous fig., from. 22 to 27, it is easy to identify the bottlenecks that occur in the used technique. The highest values imply that these procedures and their operations and computations represent the bottlenecks.

In the Initial phase, the first operation takes place which is the identifying the neighbors consumes significant time and the second operation which is labeled by the orange color in figs. 22 and 23 takes another significant time. Thus, both procedures can be seen as the bottlenecks in this state.

In the second state as in figs. 24 and 25, the second procedure, represented in the orange bar, consumes nearly more than 50% of the other operation which is the blue bar in the same figs. Lastly, in fig. 26 and 27, the minimization operation which is denoted by the blue bar takes significant time during the simulation tests. From the previous 6 figs., several factors influence and impact the system performance. Finding the nearest neighbors, calculating the weights and the number of used eigenvectors significantly play a major role in the system performance. In addition, the nature of the dataset also affects the system performance.

Table 2 displays the average spending time in every phase to compute the expected average execution time to determine the bottlenecks for unsupervised and supervised LLE.

Table 2 Average spending time in all phases in ms

State/Phase	Unsupervised LLE	Supervised LLE
	Average values	Average values
Initial	372	326
Checking	224	184
Processing	337	365

The following charts represent the actual execution time versus the predicted ones for the unsupervised and the supervised LLE in ms.

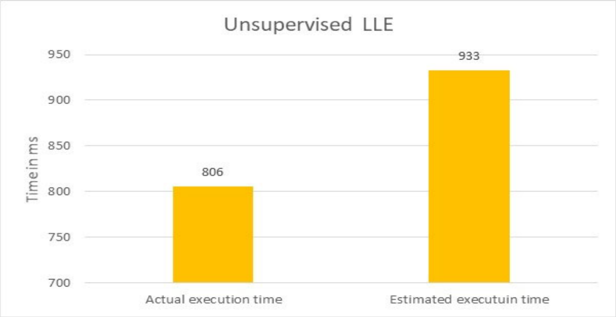

Fig. 28 Comparison between the actual and the estimated execution time for unsupervised LLE

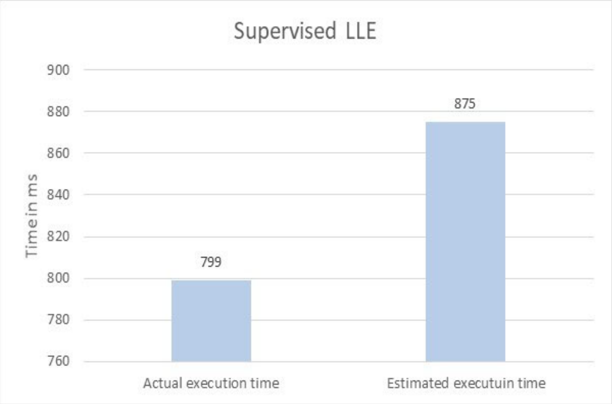

Fig. 29 Comparison between the actual and the estimated execution time for supervised LLE

Table 3 lists the error percentages between the actual values and the estimated ones for both types.

Table 3 the error percentages

Error percentage	Unsupervised LLE	Supervised LLE
	Actual vs predicted	Actual vs predicted
	13.6%	8.68%

Fig. 30 displays the error percentage between the actual and predicted values for the unsupervised and supervised LLE.

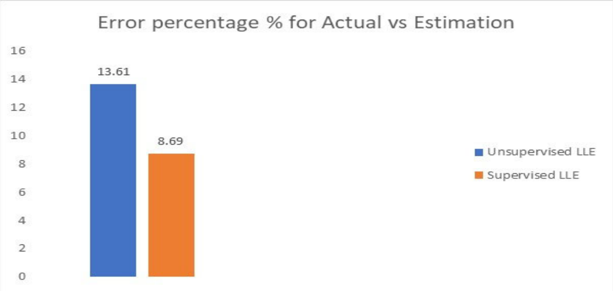

Fig. 30 The percentage of occurred error

5. Conclusion

This paper uses the developed method in [13] and [14] after adjusting it to analyze and evaluate the novel approach in [1] to spot and point the bottlenecks if exist. Figs. 22, 24 and 26 respectively clearly reveal which operations in the proposed method in [1] cause the bottlenecks. In addition, there are numerous factors that play significant attributes in inducing the execution time to be higher than what is expected. Those attributes were stated earlier in this research and in [1].

Acknowledgments

The authors gratefully acknowledge the approval and the support of this research study by the grant no. 1115-ENG-2017-1-8-F from the Deanship of Scientific Research at Northern Border University, Arar, **K.S.A.**

References

[1] M. Almutiry and A. Alsheikhy, “A Novel Method for Pattern Recognition based on Radar Tomographic Images”, IJCSNS International Journal of Computer Science and Network Security, Vol. 21, No. 10, pp. 1-12, Oct. 2021.

[2] A. Alsheikhy, “Estimating End-to-End Delay on a Networking Environment using a Developed Framework”, International journal on Information Technologies and Security (IJITS), “Officially Accepted”.

[3] A. Jastrzebska, “Time series Classification Through Visual Pattern Recognition”, Journal of King Saud University-Computer and Information Sciences, pp. 1-9, Dec. 2019.

[4] B. George, A. B. Obot and K. M. Udofia, “Modeling and Performance Evaluation of Ground based Monostatic Radar Surveillance System”, Communications on Applied Electronics (CAE), Vol. 7, No. 28, pp. 17-21, May 2019, New York, USA.

[5] E. D. Pedro, T. M. F. Alves and A. V. T. Cartaxo, “Performance Evaluation of Wavelength Division Multiplexing Photonic Analogue-to-Digital Converters for High-Resolution Radar Systems”, Optics and Photonics Journal, Vol. 9, pp. 219-243, 2019.

[6] A. Alsheikhy and M. Almutiry, “Performance Evaluation in a Radar System”, IJCSNS International Journal of Computer Science and Network Security, Vol. 18, NO. 11, pp. 116-126, Nov. 2018.

[7] D. Nagarajan and G. Kavitha, “Pattern Recognition Using Neural Network Time Series”, International Journal of Engineering and Tecknology, Vol. 7, No. 4, pp. 3357-3359, 2018.

[8] W. He, F. Ma and X. Liu, “A Recognition Approach of Radar Blips Based on Improved Fuzzy C Means”, EURASIA Journal of Mathematics Science and Technology Education, Vol. 13, No. 8, pp. 6006-6017, 2017.

[9] P. A. Waghmare and J. V. Megha, "Efficient Pattern Recognition in Time Series Data," International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, pp. 436-441, 2018.

[10] M. Almutiry, L. Lo Monte, and M. C. Wicks, “Extraction of Weak Scatterer Features Based on Multipath Exploitation in Radar Imagery,” International Journal of Antennas and Propagation, Vol. 2017. Article ID 5847872, pp. 1-13, 2017.

[11] Y. Li, X. Li, H. Wang, B. Deng and Y. Qin, “Performance Evaluation of Target Detection with a Near-Space Vehicle-Borne Radar in Blackout Condition, Journal of Sensors, Vol. 16, No. 64, pp. 1-13, 2016.

[12] A. Kumar and A. Mudaliar, “Performance Evaluation of Radar Systems”, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue, 7, pp. 6371- 6377, July 2015.

[13] A. Alsheikhy, S. Han and R. Ammar, "Hierarchical Performance Modeling of Embedded Systems“, Computers and Communication (ISCC), 2015 20th IEEE Symposium on Computers and Communications, pp. 936-942, July 2015.

[14] A. Alsheikhy, S. Han and R. Ammar, "Delay and Power Consumption Estimation in Embedded Systems Using Hierarchical Performance Modeling”, 15th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 34-39, December 2015, Abu Dhabi, UAE.

[15] A. Alsheikhy, R. Ammar, R. Elfouly and M. Alharthi, “Power Consumption and Energy Estimation in Smartphones”, International Journal of Advanved Engineering, Management and Science (IJAEMS), Vol. 2, Issue 10, pp. 1758-1765, Oct. 2016.

[16] L. Chin, .S. Wei and G. Yu, ”Performance Evaluation of Embedded System Based on Behavior Expressions”, 2nd International Conference on Mechanical and Electronics Engineering (ICMEE), Vol. 1, pp. 253-256, 2010.

[17] D. Smarkusky, R. Ammar, I. Antonios and H. Sholl, “Hierarchical Performance Modeling for Distributed System Architectures,” Computer and Communications, 2000. Proceedings. ISCC 2000. Fifth IEEE Symposium, pp. 659–664, July 2000.

[18] R. Ammar, “Performance Evaluation Analysis”, Lecture Notes, University of Connecticut, 1991.