Jordanian Journal of Computers and Information Technology

September 2021 VOLUME 07 NUMBER 03
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

PAGES PAPERS

223 - 238 A HYPER-SURFACE-BASED MODELING AND CORRECTION OF BIAS FIELD IN MR IMAGES
Daouia Azzouz and Smaine Mazouzi

239 - 252 INTRODUCING A NEW ROUTING ALGORITHM FOR WIRELESS NETWORKS ON CHIP USING REINFORCEMENT LEARNING
Zohreh Harati, Esmaeel Tahanian, Alireza Tajary and Mansoor Fateh

253 - 267 INCREASING SECURITY IN MILITARY SELF- PROTECTED SOFTWARE
Carlos M. Gonzalez

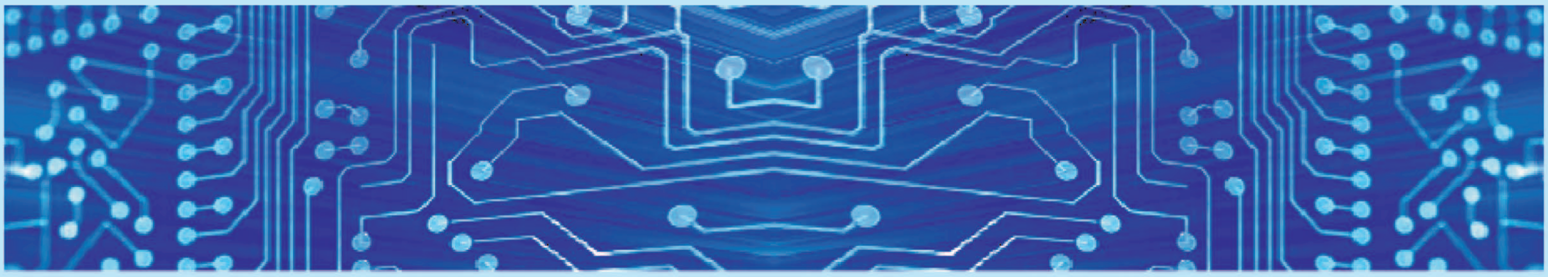
268 - 277 DECREASING THE RA COLLISION IMPACT FOR MASSIVE NB-IOT IN 5G WIRELESS NETWORKS
Bilal Rabah Al-Doori and Ahmed Zurfi

278 - 291 A NEW ADAPTED CANNY FILTER FOR EDGE DETECTION IN RANGE IMAGES
Mohamed Cheribet and Smaine Mazouzi

292 - 312 WEIGHTED GREY WOLF OPTIMIZER WITH IMPROVED CONVERGENCE RATE IN TRAINING MULTI-LAYER PERCEPTRON TO SOLVE CLASSIFICATION PROBLEMS
Alok Kumar, Lekhraj and Anoj Kumar

313 - 325 COMPARATIVE STUDY OF MACHINE LEARNING AND DEEP LEARNING ALGORITHM FOR FACE RECOGNITION
Nikita Singhal, Vaishali Ganganwar, Menka Yadav, Asha Chauhan, Mahender Jakhar and Kareena Sharma

www.jjcit.org jjcit@psut.edu.jo



An International Peer-Reviewed Scientific Journal
Financed by the Scientific Research Support Fund

Jordanian Journal of Computers and Information Technology (JJCIT)

The Jordanian Journal of Computers and Information Technology (JJCIT) is an international journal that publishes original, high-quality and cutting edge research papers on all aspects and technologies in ICT fields.

JJCIT is hosted by Princess Sumaya University for Technology (PSUT) and supported by the Scientific Research Support Fund in Jordan. Researchers have the right to read, print, distribute, search, download, copy or link to the full text of articles. JJCIT permits reproduction as long as the source is acknowledged.

Aims and Scope

The JJCIT aims to publish the most current developments in the form of original articles as well as review articles in all areas of Telecommunications, Computer Engineering and Information Technology and make them available to researchers worldwide. The JJCIT focuses on topics including, but not limited to: Computer Engineering & Communication Networks, Computer Science & Information Systems and Information Technology and Applications.

Indexing

JJCIT is indexed in:

ELSEVIER Scopus

DOAJ DIRECTORY OF OPEN ACCESS JOURNALS

crossref Text and Data Mining

OCLC WorldCat

BASE

ACADEMIA

Google scholar

JJCIT Address

Website: www.jjcit.org
Email: jjcit@psut.edu.jo
Address: Princess Sumaya University for Technology, Khalil Saket Street, Al-Jubaiha
B.O. Box: 1438 Amman 11941 Jordan
Telephone: +962-6-5359949
Fax: +962-6-7295534

Editorial Board

International Advisory Board

“Opinions or views expressed in papers published in this journal are those of the author(s) and do not necessarily reflect those of the Editorial Board, the host university or the policy of the Scientific Research Support Fund”.

"مـا ورد فـي هـذه المجلـة يعبـر عـن آراء البـاحثين ولا يعكـس بالضـرورة آراء هيئـة التحريـر أو الجامعـة أو سياسـة صندوق دعم البحث العلمي".

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 03, September 2021.

A HYPER-SURFACE-BASED MODELING AND CORRECTION OF BIAS FIELD IN MR IMAGES

Daouia Azzouz[1] and Smaine Mazouzi[2]

(Received: 29-Mar.-2021, Revised: 16-May-2021, Accepted: 25-May-2021)

ABSTRACT

Dealing with the different artifacts in medical images is necessary to perform several tasks, including segmentation. We introduce in this paper a novel method for bias field correction in Magnetic Resonance Imaging (MRI). Using the segmentation results obtained by a modified Expectation Maximization clustering, the bias field is fitted as a hyper-surface in a 4D hyper-space. Then, it is corrected based on the fact that voxels belonging to the same tissue should have the same intensity in the whole image. So, after a quick and coarse unsupervised voxel labeling by clustering by parts is performed, the bias field is computed for reliably labeled voxels. For the less reliably labeled voxels, the bias field is interpolated using a hyper-surface, estimated by a 4D Lagrangian interpolation. We evaluated the efficiency of the proposed method by comparing segmentation results with and without bias field correction. Also, we used the coefficient of variation within the MRI volume. Segmentation results and the coefficient of variation results were significantly enhanced after bias field correction by the proposed method.

KEYWORDS

MRI data labeling, Bias field correction, EM clustering, 4D hyper-surface, Lagrangian interpolation.

1. INTRODUCTION

One of the specific artifacts in Magnetic Resonance Imaging (MRI) is the Intensity Non-Uniformity (INU) across the data volume. This artifact consists in a slow and smooth variation of the intensity, whereas it should be the same for all the voxels in the same tissue. The Intensity Non-Uniformity (INU) is caused by several combined factors, the two main of which are the lack of sensitivity of the radio-frequency (RF) of the coils and the attenuation of the RF signal inside the tissues [29]. The challenge in such a problem is that it cannot be considered as a conventional additive Gaussian noise that can be efficiently removed by denoising methods. So, the intensity non-uniformity, produced as a bias field, is a full hard problem and thus it requires specific methods for estimation and correction. Methods for image segmentation or image registration must take into account the bias field and deal with it in order to provide reliable processing results. Some of these methods proceed by jointly, correcting the bias field and segmenting or registering images [4], [27], [6], [8], [13]. Nevertheless, most of them consider the bias field correction as a pre-processing task that precedes the segmentation or the registration [5], [16], [13]. The hardness of the problem lies in the fact that the intensity variation caused by the bias field is very slow and so, it is hard to detect locally.

We propose in this paper a novel method for bias field correction in MR images, based on a quick and coarse segmentation of MR data. Using the resulting segmentation, a hyper-surface in a 4D space, that models the bias field in the whole 3D image volume, is fitted using a Lagrangian interpolation. First, a modified EM (Expectation Maximization)-based clustering is performed on a set of sub-volumes, covering the entire volume of the image. So, the obtained labeling results in the different sub-volumes are merged in order to obtain the segmentation of the whole volume. Such an EM-clustering by parts allows to reduce false voxel labeling that occurs when the clustering is performed in the whole volume. The labeling in the whole volume results in three sets of voxels, corresponding respectively to the three tissues of interest of the brain matter; namely: Cerebro-Spinal Fluid (CSF), Gray Matter (GM) and White Matter (WM). Then, only the voxels with high membership certainty (most confidently labeled) are used to estimate the bias field at their respective positions in the image. Using the estimated bias field values at the reliably labeled voxels, a 4D hyper-surface is fitted in the 4D hyper-space (X,Y,Z,I), using a 4D Lagrangian surface [38]. Based on the fitted hyper-surface, the bias field is calculated for the remainder of the less reliably labeled voxels and so, the intensity can be corrected in the whole image. According

D. Azzouz[1] and S. Mazouzi[2] are with University 20 Aout 1955, Skikda, Alegria. Emails: [1]azzouz.daouia@gmail.com, [1]d.azzouz@univ-skikda.dz and [2]s.mazouzi@univ.skikda.dz.

to the literature, polynomial fitting was used several times to model the bias field [36], [34], [19]. However, on one hand, if the used polynomials are with low order, they do not well fit the bias field [19]. On the other hand, if the order of the used polynomials is high, such in polynomials, computation results in a combinatorial explosion of the number of parameters and therefore a prior knowledge must be provided in order to use low-order Legendre polynomials [34]. Unlike the state-of-the-art methods that use highly computational polynomials for the bias field surface fitting, our method is based on Lagrangian polynomial, where the order can be high, nevertheless the number of the involved parameters remains low.

Evaluating the performances of a method for bias field correction can be performed by comparing the segmentation results without and with bias correction, by using the same method of image segmentation [9]. It has been obtained that the bias field estimation and correction according to our method, using the proposed modified EM clustering algorithm, allows to significantly enhance the segmentation of MR images. Furthermore, the coefficient of variation (*CV*) has been used to show that the intensity homogeneity was enhanced for both the simulated and the real MRI involved in the experimentation.

The remainder of the paper is organized as follows: In Section 2, we introduce a short review of some well-referenced works in the literature, having dealt exclusively or jointly with bias field correction. Section 3 is devoted to the new method proposed in this work, where we show how a prior fast and coarse segmentation is performed using a modified EM-based clustering by parts, then how the bias field is firstly calculated for the reliably labeled voxels. In the remainder of the same section, we show how the bias field is fitted using a hyper-surface and how the image intensity is corrected. Section 4 is reserved for the experimentation of the proposed method, where we show and compare the segmentation results with and without bias field correction. Furthermore, the results of the coefficient of variation are computed and compared with those of other methods. It will be noted that the intensity was significantly enhanced in the involved MRI. In Section 5, we summarize the work and highlight some of its potential perspectives.

2. RELATED WORK

Several authors have proposed different methods for bias field correction in MR images. Some methods, in particular the earlier ones, were filtering-based [36]. For such methods, the bias field is removed by considering it as a low-frequency artifact, compared to high-frequency anatomic structures [24]-[25]. Filtering-based methods suffer from two common problems that lead to less accurate bias field removed and therefore to less efficient image post-processing: First, they significantly alter the image near the edges of the regions (Hallo effect), so that a special processing should be considered to preserve edge sharpness [2], [18]. Second, they erase low-frequency structures as a result of spatial blind filtering. To deal with this problem, some authors, such as those in [21], were able to preserve low-frequency structures by a combination of singularity functions. Following the same approach, the authors in [33] have applied homomorphism filtering for bias-field correction. Their method consists in extracting the log-bias by low-pass filtering, then the latter is subtracted from the log-image in order to obtain the log-corrected image. Aiming at preserving image details while the bias field is corrected, the authors in [8] remove the bias field by extracting image details after multi-layered Gaussian filtering.

Level set formulation was used by Li et al. for jointly correcting the bias field and segmenting the image [20]. By minimizing the level set functional that defines the clustering criterion, the segmentation of the image and the correction of the bias field are jointly performed. In another work, Chang et al. have proposed a new variation model for INU correction for Rodent Brain MRIs [7]. Based on Mumford–Shah functional, the authors define several terms to be minimized, aiming at extracting the bias mask. A morphological processing is also applied in order to enhance the obtained mask. Based on local and global information, Cong et al. proposed a novel model for MRI segmentation and INU correction [10]. Neighborhood information defines local constraints and the global regularization is performed based on global spatial information. In a recent work, Shan et al. [28] proposed a region-based active contour model with interleaved image segmentation and INU correction. A global energy functional, which was obtained by combining local energies, is fitted thanks to a level set method, where two regularization terms allow to fit the image energy functional.

Other authors have assumed that the intensity of a given tissue does not change in the whole image unless because of the bias field. So, the bias field is estimated as the intensity variation within the same type of tissue. This variation is often represented by a surface that must be fitted, often by B-spline approximations. In [22], the authors use a second-order polynomial to model the intensity within a dominant tissue, automatically extracted. A Gaussian model of the dominant tissue was beforehand proposed in [35]. The main drawback of such methods is that they were designed for a dominant tissue. For the other tissues, the bias field is not or at most roughly estimated. In [32], the bias field was taken into account in the similarity measurement that allows to enhance the separability of MRI data. Moreover, the formulated objective function of the Fuzzy C-Means (FCM) algorithm integrates a regularization term that compensates the bias field. Meena Prakash et al. have combined the FCM algorithm and Expectation-maximization (EM) algorithm for brain MRI segmentation with bias correction [26]. In their implementation, they take into account the spatial information, by incorporating it by convolution of the posterior probability during E-Step of the EM algorithm. Mishro et al. have also opted for fuzzy clustering for INU correction in brain MRIs [23]. They also incorporate spatial information by altering the membership matrix of standard FCM, aiming to attenuate the effect of noise and INU. So, equidistant pixels are assigned to a single cluster.

Recently, machine leaning-based methods for INU correction started to be proposed. Dai et al. [11] proposed a deep learning-based INU correction algorithm, called residual cycle generative adversarial network (res-cycle GAN), which consists of the calculation of the inverse transformation between the INU uncorrected and corrected MRIs; so it will be possible to compute the INU corrected MRIs.

3. Prior Segmentation and Bias Field Estimation and Correction

We remind that the bias field, which consists in a non-uniformity of the intensity of the magnetic field, causes a slow and smooth variation of the luminance within MR images. This artifact leads to erroneous results of image segmentation, because voxels belonging to a same tissue could have significantly different intensities, especially if they are far from each other (see Figure 1). In order to correct the intensity in the raw image, the bias field must be estimated. In this work, a 4D hyper-surface is fitted in order to model the bias field in the whole 3D image. The principle of the proposed method consists to asses that in a small region, after it is smoothed to reduce the noise within, voxels should have the same average intensity if they belong to the same type of tissue. According to this principle, we proceed by segmenting small sub-volumes in the MRI, where we can assume that the intensity variation due to the bias field can be neglected. Then, the segmentation of the whole MRI is obtained by the fusion of the different sub-segmentations. So, as the segmentation is assumed correct, the bias field at a given voxel is expressed by the ratio between the intensity of the voxel and the intensity of a selected voxel, called voxel of reference that belongs to the same tissue as the voxel in question. Such an assumption is based on the following model that expresses how the true image I was altered by the bias field β and affected by a Gaussian noise η, resulting in the measured image $\hat{I}$.

$$I(x,y,z) = (I(x,y,z) + \eta(x,y,z)) \times \beta(x,y,z) \tag{1}$$

So, the main stages for bias estimation and intensity correction according to our method are as follows: First, a voxel labeling is performed by an EM-based clustering algorithm that is executed separately on several sub-volumes. Then, the results of voxel labeling in the different sub-volumes are merged. Such a local clustering avoids to gather voxels that belong to a given same tissue, but have different intensities because they are far from each other. Note that a global clustering that involves the whole 3D volume results in a high rate of erroneous voxel labeling, in particular when the bias field level is high (see Figure 1). So, we introduce in sub-section 3.1 a new iterative EM-based algorithm for voxel labeling. The algorithm generates random sub-volumes within the MRI volume, where the bias field at each sub-volume is low, what allows sufficiently reliable voxel labeling. After the voxels in the whole volume are labeled according to a given merge scheme, the bias field is estimated first for the set of the voxels that are reliably labeled, according to their membership certainty. Then, for each connected set of less-reliably labeled voxels, where the bias was not calculated, a cuboid is set, in which a 4D hyper-surface will be fitted. Control points used to fit the hyper-surface are the voxels in the cuboid where the bias field was calculated. Finally, the bias field at the less reliably labeled voxels in the cuboid is interpolated using the fitted hyper-surface (see Figure 2).

3.1 Voxel Labeling by Iterative EM Clustering by Parts

We assume that MR images are skull-striped, using a brain extractor tool, such as BET of FSL [30],

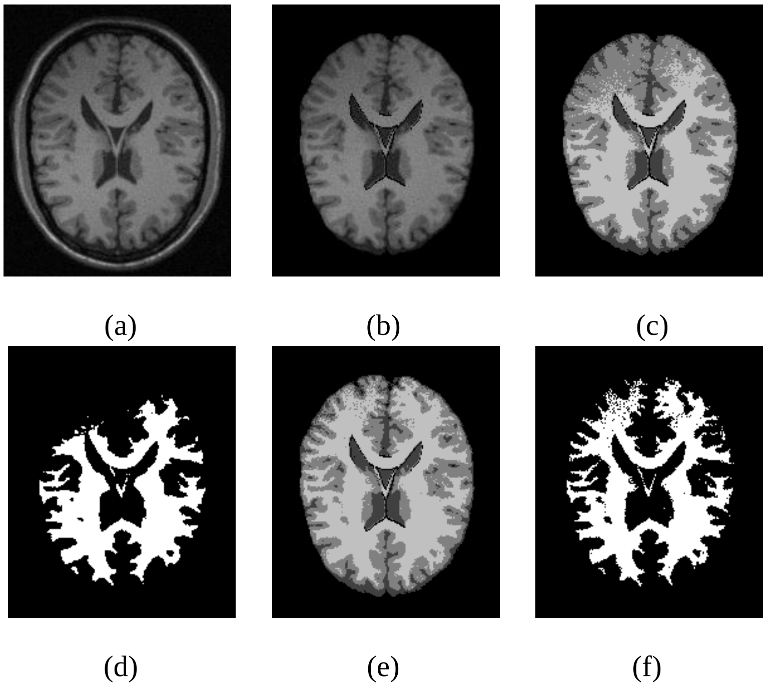


Figure 1. Erroneous labeling with presence of high INU: a) Raw MRI with 3% noise level and 90% INU level; b) Extracted brain tissues; c) Global EM clustering results, where we can notice the strong alteration of the white matter at the top of the image which was erroneously labeled as gray matter; d) Extracted white matter, where an important part at the top is truncated; e) Voxel labeling by the modified EM by parts algorithm; f) Resulting white matter.

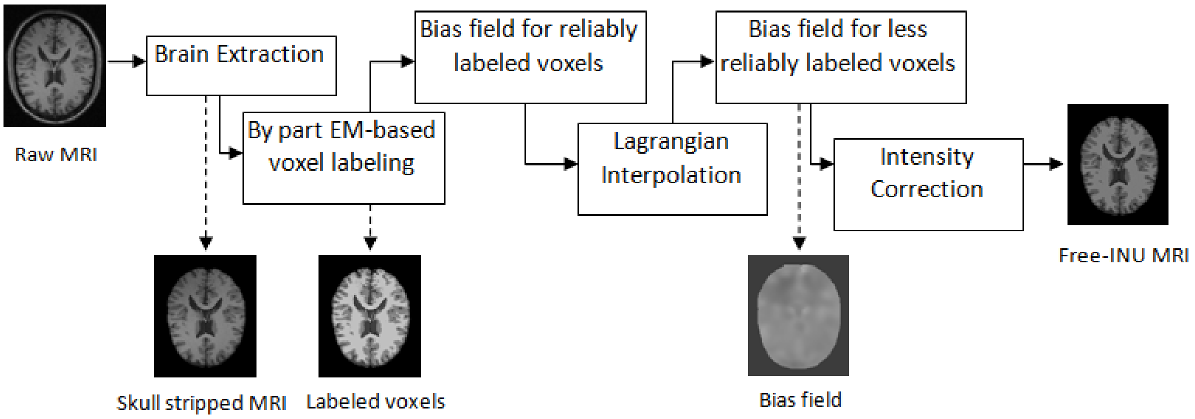


Figure 2. Processing steps according to the proposed method for bias field estimation and INU correction.

where only the brain tissues remain in the image; namely: CSF (Cerebro Spinal Fluid), GM (Gray Matter) and WM (White Matter). Considering the model for image formation cited above, it is not required that the image be denoised in order to perform the correction of the bias field. Indeed, the used bias represents the ratio between the noised measured image and the noised bias-free one (see Formula 1). Nevertheless, after the bias is corrected, it is suitable to denoise the image after assuming a noise model or by jointly image denoising and voxel labeling, as done in several works [12], given that prior global denoising could aggravate the partial volume effect problem, by which MR images are affected [2]. However, this issue is not included in our interest in this work.

By considering that MR data follows a Gaussian Mixture Model (GMM), the EM algorithm is well suited for both the estimation of the distribution parameters and image segmentation by voxel labeling. By taking into account the particularity of the bias field, that produces significantly different intensities for the set of voxels that belong to a same tissue, the EM clustering is performed for a set of sub-volumes, where the bias field is sufficiently weak in each sub-volume and then the resulting partitions are merged. In a sub-volume, noted *s*, there should exist the 3 classes of the brain tissues; namely, CSF, GM and

WM. The parameters to estimate by the EM algorithm for each tissue class *T* (*T* = 1, 2, 3) in the sub-volume *s* are the mean intensity $\mu^{s}{}_{T}$ and the standard-deviation $\sigma^{s}{}_{T}$. The probability density, assuming a Gaussian mixture, is given for a tissue class (*T*), by:

$$f_T(x_i;\ \mu_T^S, \sigma_T^S) = \frac{1}{\sigma_T^S\sqrt{2\pi}} e^{-\frac{1}{2}(x_i-\mu_T^S)/2{\sigma_T^S}^2} \tag{2}$$

According to this model, the adequacy of the voxel *i* to the tissue class *T*, which we consider as the membership certainty, is expressed as follows:

$$p_i^T = \frac{\pi_T f_c(x_i;\mu_T^S,\sigma_T^S)}{\sum_{j=1}^{k} \pi_T f_T(\bar{x}_j;\mu_j^S,\sigma_j^S)} \tag{3}$$

By using the adequacy probability, the distribution parameters are readjusted as follows:

$$\mu_T^S = \frac{1}{n_T^S}\sum_i p_i^T x_i \tag{4}$$

$$\sigma_T^S = \frac{1}{n_T^S}\sum_\imath p_\imath^T (x_\imath - \mu_T^S)^2 \tag{5}$$

$$\pi_T = \frac{n_T^S}{n^S} \tag{6}$$

where n^{s} and $n^{s}{}_{T}$ are respectively the size of the set of all the voxels in the considered sub-volume *s* and the size of the cluster corresponding to the tissue class *T* in the same sub-volume.

To insure label integrity across the different sub-volumes, each one of the latter must contain all the considered labels (CSF, GM and WM). So, a sub-volume is set randomly within the global volume and after the EM clustering by parts is executed, the sub-volume will be retained only if the 3 labels are present. Otherwise, the sub-volume is rejected and another is randomly reset. As sub-volumes are randomly selected, a given voxel can be labeled several times according to the different sub-volumes to which it belongs. Also, the labels assigned to a same voxel may be different from a sub-volume to another. So, we calculated, for each voxel, the occurrences of labels assigned to it during sub-clustering that include the voxel. At the end of the clustering iterations, the label retained for a given voxel is that having maximum number of occurrences.

The following algorithm shows how the iterative clustering by parts is performed. We have opted for an

Algorithm 1 *Iterative EM Clustering by Parts()*

```
Inputs:    Skull stripped MRI volume
Outputs:  Labels
for every voxel v in the MRI volume do
     for every label l do
          LabelOccurencies[v][l] ← 0
     end for
end for
NumberIterations ← 0
repeat
     repeat
          Subvolume ← random subvolume (xt, yt, zt, xb, yb, zb)
          Perform EM Clustering in Subvolume
     until Number of labels in Subvolume = 4 // background included
     for every voxel v in Subvolume do
          l ← label of v
          LabelOccurencies[v][l]++
     end for
   NumberIterations++
until NumberIterations = maxIterations
for every voxel v in the MRI volume do
          Labels[v] ← ArgMax_l(Label Occurencies[v][l], l ∈ [0..3]) // 0: background
end for
```

EM-based clustering instead of *k*-means one, because the former provides for every voxel the membership certainty to a given cluster. This certainty allows us to select only the voxels that certainly belong to their respective tissues. *K*-means clustering, which does not use a mixture of distributions, cannot quantify the membership certainty to the clusters and so does not allow to select reliably labeled voxels to fit the bias field.

As a result of this first stage, we obtain a segmentation of the image, where each voxel is assigned to a tissue class with its membership certainty. For each tissue class, the voxels that will be used to estimate the bias field are those with the membership certainty greater than a given threshold T_{p}, which will be set experimentally by using a set of MRIs with their respective group truth segmentation (see Section 4).

3.2 Bias Field Estimation and Intensity Correction

The intensity non-uniformity of the magnetic field leads to a disparity of the intensity values of the voxels belonging to the same tissue. It consists of a slow variation of the intensity according to an unknown model. So, the latter must be estimated in order that intensities can be corrected. In our work, we compute an initial voxel labeling in order to estimate the bias field and then correct the intensities. So, a coarse segmentation of the image is performed using an EM algorithm executed by parts in the global volume. After merging partial labeling results and the final labels affected to the image voxels, the bias field is estimated as a hyper-surface *β*, where *β(x,y,z)* expresses the ratio between the mean intensity around the voxel (x,y,z) and the mean intensity at a voxel of reference $(x_{r,},y_{r},,z_r)$ belonging to the same type of tissue and having the best membership certainty among all the voxels in the current sub-volume. We consider a neighborhood of 3×3 voxels around a given voxel to calculate the mean intensity value.

$$\beta(x,y,z) = \frac{\tilde{\hat{I}}(x,y,z)}{\tilde{\hat{I}}(x^r,y^r,z^r)} \tag{7}$$

where,

$$\tilde{\hat{I}}(x,y,z) = \frac{1}{|\chi(x,y,z)|}\sum\nolimits_{i\in\chi(x_i,y_i,z_i)} \hat{I}(x_i,y_i,z_i) \tag{8}$$

where $\chi(x_i, y_i. z_i)$ is the set of neighboring voxels $\{(x_i, y_i, z_i)\}$ of (x, y, z) that belong to the same tissue.

For the previous treatment, we consider only the voxels $\{i\}$ that certainly belong to their respective tissue classes. A voxel i is retained for bias estimation, if its corresponding membership certainty, expressed by the probability $p^T{}_i$ is greater than a given threshold T_p. For the rest of the voxels, the bias field is computed by interpolation using a Lagrangian hyper-surface [38] that we introduce in this work. The hyper-surface is computed using a set of voxels, sampled from those for which the bias field has been estimated. For each set of connected unlabeled voxels, for which we aim at computing the bias field, an including volume area is defined for this set. Within this volume, a set Ω of approximatively uniformly distributed labeled voxels are selected (see Figure 3(c)). This set is used as the control points to generate the Lagrangian hyper-surface, estimating the bias field in this area (see Figure 3(d)).

The Lagrangian interpolated hyper-surface, $\hat{\beta}(x,y,z)$ in an including volume area is expressed as follows:

$$\hat{\beta}(x,y,z) = \sum_{k\in\Omega} \beta_k \frac{\prod_{i\neq k}(x-x_i)(y-y_i)(z-z_i)}{\prod_{i\neq k}(x_k-x_i)(y_k-y_i)(z_k-x_i)} \tag{9}$$

where $\{(x_k, y_k, z_k) \in \Omega\}$ is the set of control points in the including volume area used to express the Lagrangian polynomial. Furthermore, a given point (x_k, y_k, z_k) is retained as a control point only if: first it was labeled belonging to one tissue and second its membership certainty is greater than the threshold T_{p}. Once the bias field is computed in the whole MRI volume, the voxel intensity $\widehat{I^c}(x,y,z)$ at every voxel (x,y,z) of the volume is corrected as follows:

$$\hat{I}^c(x,y,z) = \hat{I}(x,y,z) \times \hat{\beta}(x,y,z) \tag{10}$$

The obtained corrected image $\widehat{I^c}$ is considered bias field-free and it can be used for further processing, such as more accurate image segmentation or image registration.

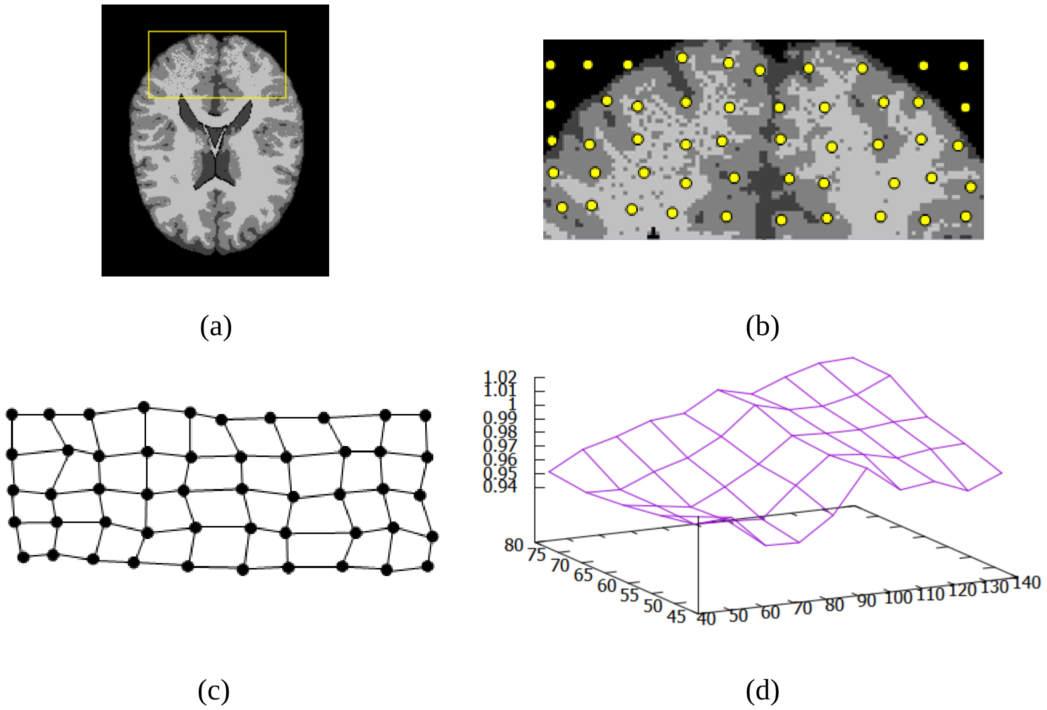
Contrary to non-interpolation-based methods, such as B-spline-based ones [14], the proposed INU correction based on Lagrangian interpolation allows to preserve the values of the bias field at the reliably labeled voxels, considering that the latter are not affected by other artifacts, such as noise or partial volume effect. For less reliably labeled voxels, the bias field is estimated by the Lagrangian interpolation, resulting in a value better than the one directly calculated (according to Formula 7).

The overall proposed method can be expressed according to the following algorithm:

Algorithm 2 *INU Correction()*

```
Inputs:  MRI volume
Outputs:  Free-INU Skull stripped MRI volume

MRI Skull Stripping by BET
Labels ← EM-Clustering-by part (Skull Stripped MRI)
for each connected part CP of homogeneous voxels in Skull stripped MRI do
   $Set_1$ ← Set of reliably labled voxels of CP
   $Set_2$ ← Set of less reliably labeled voxels of  CP // $Set_2$ =  CP-$Set_1$
   Compute the bias field B according formula 7 for the voxels in $Set_1$
   Compute Lagrangian polynome using bias fiels of all connected parts of $Set_1$  (formula 9)
   Estimate bias field of $set_2$ using the computed Langrangian polynomial (using formula 9)
end For
Correct the MRI INU for all the voxels by formula 10
```



(a)

(b)

(c)

(d)

Figure 3. Selection of control points for fitting the hyper-surface: a) Slice from a clustered MRI, b) Point selection on 2D grid: for each regular position on the grid, the best voxel in the neighborhood are selected, having the highest membership certainty, c) Control points for Lagrangian interpolation, d) Bias field surface. The 3D surface is the projection of the 4D surface on the space (*X*,*Y*,*I*) corresponding to the 2D considered slice.

4. EXPERIMENTATION

We have experimented our method on simulated MRIs from the Brain Web database (https://brainweb.bic.mni.mcgill.ca) and on real MRIs from the Internet Brain Segmentation Repository IBSR database (https://www.nitrc.org/projects/ibsr). Simulated MRIs from the Brain Web were widely used in similar works, given that the Brain Web platform allows to customize artifact levels, in particular

INU. All used simulated MRIs are 181×217×181 voxels of size. Brain Web provides also ground-truth segmentation that allows us quantifying the efficiency of our method by comparing the segmentation results before and after bias field removal. We have considered several MRIs all with T1 modality. The INU in the considered MRIs ranges in {30%, 40% and 70%} and noise in {3% and 5%}. For real MRIs from IBSR database, all the 18 skull-stripped MRIs were considered. Each IBSR MRI is a volume of 256×256×128 voxels. We quantify the efficiency of the intensity correction by two different methods: First, by measuring the quality of the MRI segmentation by referring to its ground-truth segmentation provided by the used databases. Second, by measuring the coefficient of variation obtained for each image before and after bias field removal. For the first method, we use the overlap index kappa (κ). For a given tissue class T, it is expressed as: $\kappa_T = 2\times TP_T / (2\times TP_T + FP_T + FN_T)$, where TP_T , FP_T and FN_T are the numbers of respectively, correctly labeled voxels (true positives), voxels wrongly labeled as belonging to the tissue class T (false positives) and voxels wrongly labeled not belonging to the tissue class T (false negatives). Dice index expresses the overlap between the resulting segmentation and the ground truth segmentation, which expresses in turn several aspects, including accuracy. For the second method, the coefficient of variation is calculated as follows: $CV_T = \sigma_T/\mu_T$. It allows to estimate the variation amount around the mean value. When CV_T is low, the intensity non-uniformity is well corrected for the tissue class T.

4.1 Threshold Initialization

Only one threshold, T_p is used for our method for bias field estimation that we initialize experimentally by using a set of MRIs with their ground truth segmentation, as a learning set. The latter is composed of 6 MRIs, obtained by varying the couple (N%, INU%) in {1,3,5} x {20,60}. The optimal value of T_p corresponds to the maximum of the κ averages for the gray matter in the whole learning set.

Table 1. Initialization of the threshold T_p according to κ index for the training MR images.

$\boldsymbol{T_p}$	0,50	0,55	0,60	0,65	0,70	0,75	0,80	0,85	0,90	0,95
$\boldsymbol{\kappa}$	0,84	0,85	0,87	0,91	0,91	0,92	0,92	0,90	0,90	0,84

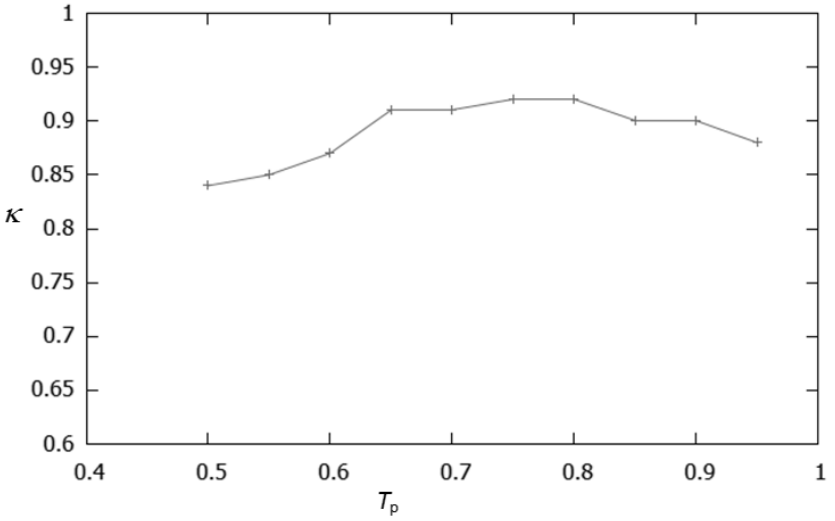
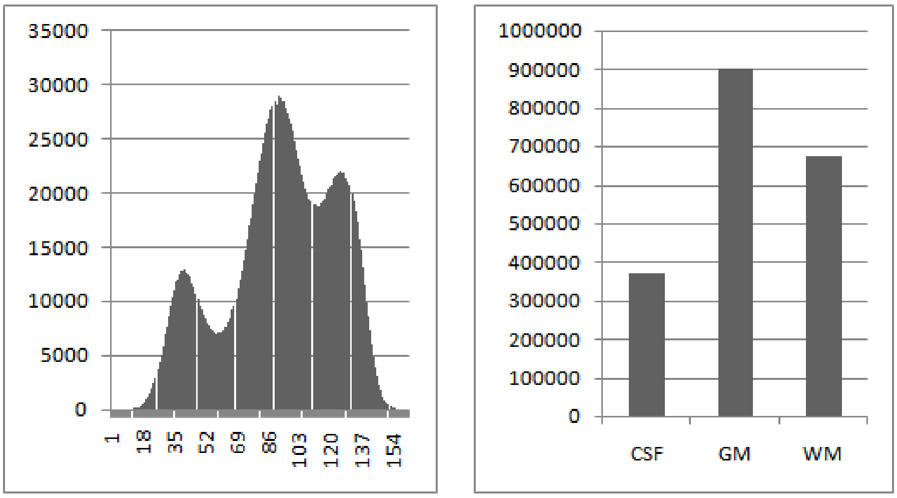


Figure 4. Average κ index according T_p threshold.

Table 1 and Figure 4 show the variation of the κ average for the gray matter. According the obtained results, T_p is set to 0.775.

4.2 Experimental Results

We consider only MRIs corresponding to healthy subjects, where the aim is to correct the INU after segmentation by extracting the three MRI tissues (CSF, GM and WM). To perform fully automatic segmentation, we should automatically set the tissue class T according to the tissue in which we have interest. To do that, we take into account the knowledge that the cluster of the gray matter is the largest one and the cerebrospinal fluid is the smallest one (see Figure 5). The white matter cluster is in the middle [1].



(a) (b)

Figure 5. Distributions of the intensities of an MRI: a) For a raw MRI : modes of the graphic, starting from left, correspond respectively to CSF, GM and WM, b) Distribution of the labels in the corresponding ground truth segmentation.

4.2.1 INU Correction Evaluation by Segmentation Comparison

Figure 6 shows the steps of the bias field correction and the obtained labeling of the tissues of interest: CSF, GM and WM. The MRI was generated by setting the noise level N% to 3 and the bias field INU% to 50. By observing images of EM clustering by parts (see sub-section 3.1), before and after bias field correction, we can see the set of voxels where the labels were corrected, in particular voxels at the top of the image after bias correction. The bias field in Figure 7 is a projection of the hyper-surface on the 3D space (*X*, *Y*, *I*), resulting in a conventional 3D surface.

Table 2. Dice coefficient for several MRIs obtained by different combinations of noise N% and intensity non-uniformity INU% before and after bias field correction.

		After bias correction		**Before bias correction**	
		N%=3	**N%=5**	**N%=3**	**N%=5**
INU%=30	Dice(WM)	93,87	92,60	90,82	88,43
	Dice(GM)	92,01	90,58	87,14	84,21
	Dice(CSF)	86,74	84,73	81,52	80,81
INU%=40	Dice(WM)	92,85	92,46	89,49	87,73
	Dice(GM)	92,01	90,58	87,14	84,21
	Dice(CSF)	84,21	82,76	79,63	78,93
INU%=70	Dice(WM)	91,52	90,94	87,34	85,15
	Dice(GM)	90,82	89,59	86,38	85,13
	Dice(CSF)	83,64	81,19	77,14	74,24

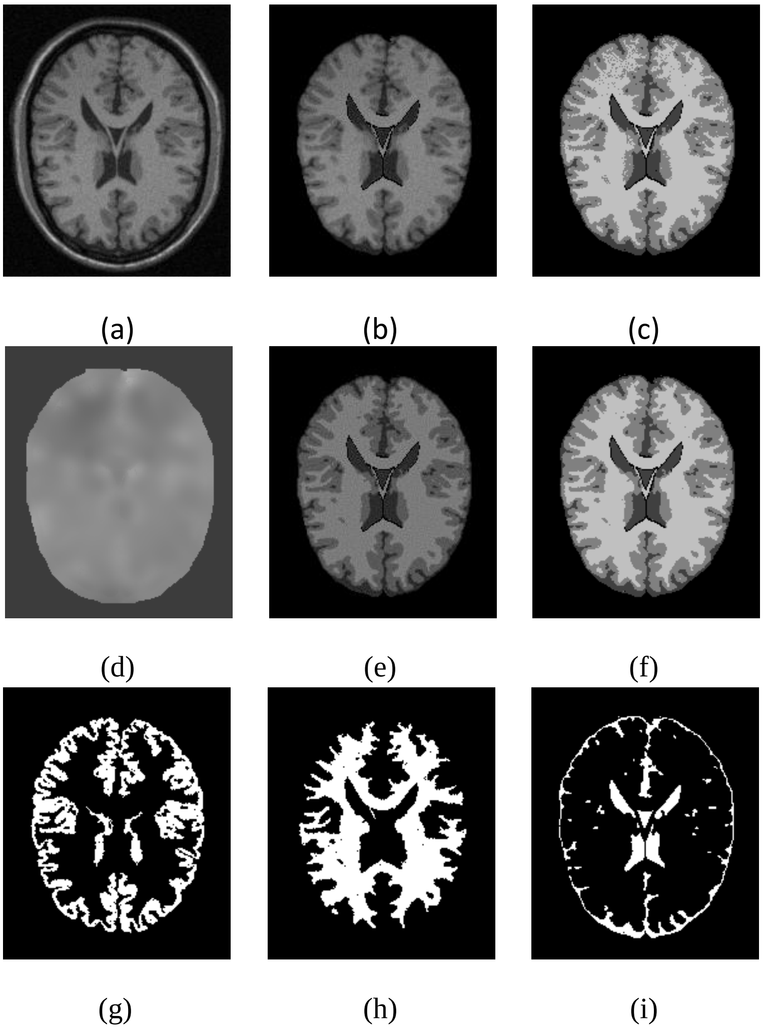


Figure 6. Steps for bias field correction : a) Raw MRI (before skull-stripping), b) Skull-stripped MRI, c) EM clustering before bias field correction, d) A slice from the bias field (projection) e) Brain after bias correction, f) EM clustering after bias field correction, g) Extracted GM, h) Extracted WM and i) Extracted CSF.

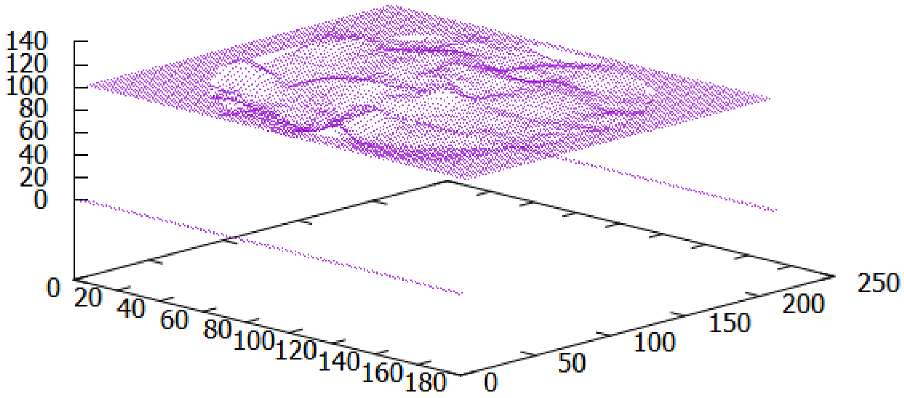
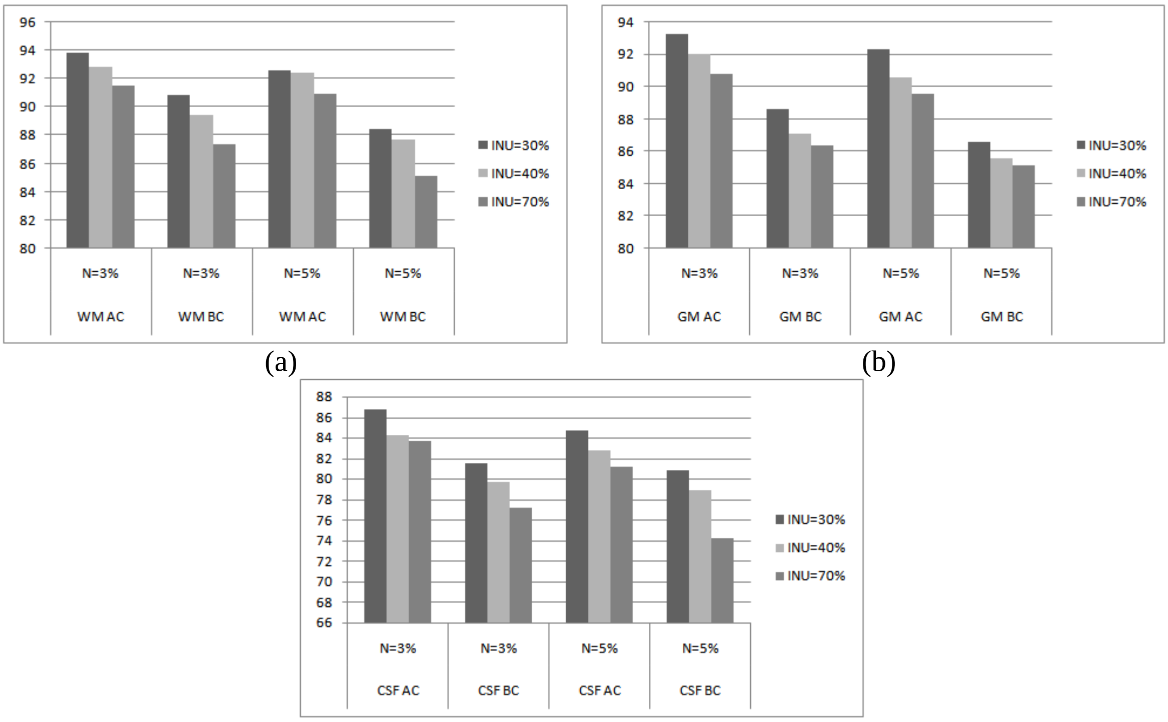


Figure 7. Projection of the bias field in the 3D space (*X*, *Y*, *I*) corresponding to a slice for which *Z* is fixed.

Table 2 and Figure 8 show how the bias field estimation and correction have allowed to enhance the labeling of the voxels in the different tissues. We can notice that for different combinations of noise level and INU level, the intensity disparity was corrected in the images even with high values of noise and INU, leading to better Dice coefficients.



(a)

(b)

(c)

Figure 8. Dice coefficient according the different tissues and according to several artifact combinations (N% and INU%): a) WM before correction (BC) and after correction (AC) , b) GM (BC and AC) and c) CSF (BC and AC). For all the combinations, voxel labeling was significantly enhanced after the bias field was corrected.

4.2.2 Evaluation Using the Coefficient of Variation

Coefficient of variation is an indirect evaluation of performance of bias field correction [9], expressed as the fraction of the mean intensity μ_T of a given tissue class *T* to its standard deviation σ_T. For a tissue class *T*, it is expressed as follows:

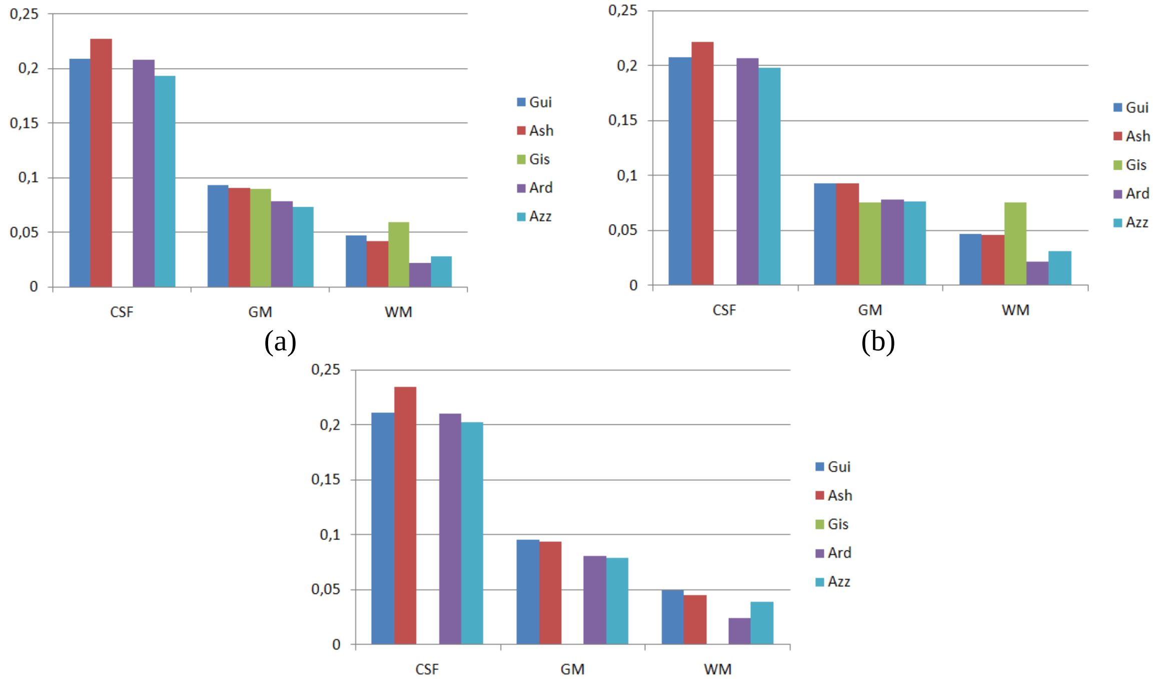
$$CV_T = \frac{\sigma_T}{\mu_T}$$

Weak values of *CV* indicate a low bias field. So, when *CV* significantly decreases after bias field correction, that reflects a good performance of the used method. Furthermore, the coefficient of variation can be used as a metric to compare bias field correction methods.

Obtained results expressed by the coefficient of variation were compared to some well-referenced work in the literature. These are the works respectively of Ashburner and Friston [3], Guillemaud et al. [17], Gisper et al. [15] and Ardizzone et al. [2]. In the first work, the authors relied on surface fitting of the bias field after considering image data as a mixture model. In the second work, the authors combine homomorphism filtering and normalize convolution after operating a coarse segmentation to separate tissues from the image background. Gisper et al. presented a locally adaptive algorithm based on the minimization of the Classification Error Rate (CER) between different cerebral tissues. Finally, Ardizzone et al. used Homomorphic Unsharp Masking (HUM), which is a filtering technique for INU correction, without producing the Halo around the edges, resulting in a HUM-based halo compensation (HC-HUM). For comparison and as for the involved authors, we have considered different INU levels, which are respectively 20%, 40% and 70%. Table 3 and Figure 9 show the ranges of the coefficient of variation for the methods involved in the comparison and for the different levels of INU. For the different tissues (CSF, GM and WM) and for the different considered INU levels (20, 40 and 70%), the proposed method (Azz.) performs better than most of the methods involved in the comparison. Indeed, it performs better than the methods of respectively Ashburner et al., Guillemaud et al. and Gisper et al. for all the tissues and all the involved INU levels. Also, the proposed method performs better than that of Ardizzone et al. in 66% of cases (tissues and INU). We note here that using simulated MRIs from the Brain Web dataset, as most of the works which have dealt with the bias field in MRIs, allows well understanding how MRIs are affected by the bias field and showing how bias field correction methods operate according to different levels of artifacts, in particular the bias field.

Table 3. Coefficient of variation obtained by the involved methods in the comparison, for different levels of INU.

	INU level								
	20%			**40%**			**70%**		
Method	**CFS**	**GM**	**WM**	**CFS**	**GM**	**WM**	**CFS**	**GM**	**WM**
Gui. [17]	0,209	0,093	0,047	0,208	0,093	0,047	0,212	0,096	0,050
Ash. [3]	0,227	0,091	0,042	0,229	0,091	0,043	0,235	0,094	0,045
Gis. [15]	-	0,090	0,059	-	0,076	0,076	-	-	-
Ard. [2]	0,208	0,078	**0,022**	0,207	0,078	**0,022**	0,211	0,081	**0,024**
Azz. (proposed)	**0,193**	**0,073**	0,028	**0,195**	**0,075**	0,031	**0,203**	**0,079**	0,039



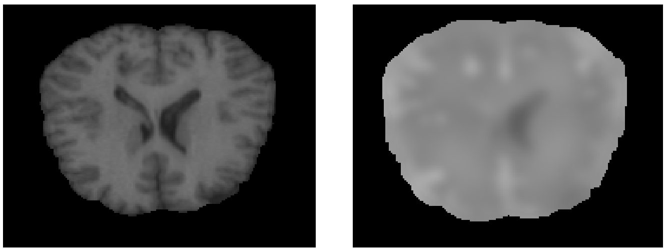
(a)

(b)

(c)

Figure 9. Coefficient of variation comparison for different levels of INU.

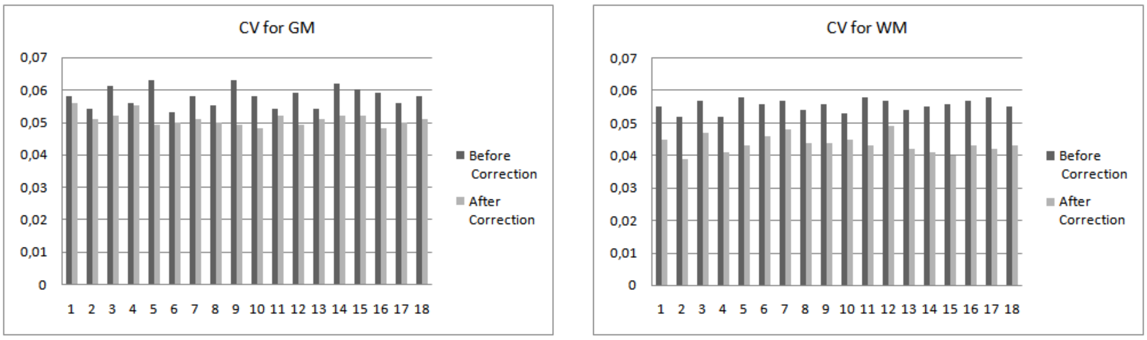
Considering real MRIs from IBSR, results of bias field correction, expressed by the coefficient of variation, were significantly enhanced. Figure 10 shows a sample of an IBSR MRI and the computed bias field. Figure 11 shows the CV values for the 18 skull-stripped MRIs. Obtained results for the 18 real MRIs, expressed by CV before and after correction, show that the bias field was significantly corrected for the whole images.



(a)

(b)

Figure 10. Bias field in a sample of real IBSR MRI: a) Raw MRI, b) Computed bias field.



(a) (b)

Figure 11. CV with IBSR: a) Average CV before correction for the 18 MRIs of the IBSR dataset, b) Average CV after correction.

4.3 Analysis of Results and Discussion

Obtained segmentation results of noisy MRIs affected by different levels of intensity non-uniformity were significantly enhanced by the proposed method. In this work, we have opted for EM clustering performed by parts in several sub-volumes, in order to produce a fast and reliable labeling of voxels. However, any method that produces voxel labeling according to the anatomical tissues can be used on condition that it should define a membership certainty for voxel labeling. With the modified EM clustering algorithm proposed in this work, the remaining voxels for which the membership certainty is below the threshold T_p and the bias field was not initially estimated were not numerous. For such sets of voxels, it was always possible to define an including sub-volume that contains sufficiently voxels for which the bias field was calculated. The latter are used as control points for fitting a hyper-surface that will be used later to interpolate the bias field at the voxels less-reliably labeled. To contrast our method, we remind that most of the proposed methods for bias field modeling and intensity correction are Bayesian- or Markovian-based [36], [31]. They proceed by simultaneously compute the restored image I, the bias field β and the noise η, considering different priors. Others are optimization-based; such as those using some objective functions, expressing in most of the cases an energy function that must be minimized [28],[37]. For our case, we have opted for a first fast voxel labeling method that produces coarse image segmentation. The latter is used to estimate and correct the INU, then the final voxel labeling is performed using the corrected data. Furthermore, using high-reliably labeled voxels in order to fit the bias hyper-surface has allowed to obtain accurate interpolation for the voxels that are less-reliably labeled.

5. Conclusion

We have introduced in this paper a novel method for bias field estimation and correction in MR images. Computing the bias field is necessary to obtain enhanced segmentation results for such images. Contrary to B-spline approximation, we have opted for Lagrangian interpolation to estimate the bias field in the whole 3D data volume as a hyper-surface in a 4D space. Such a choice allows to accurately compute the bias field at the voxels that are reliably labeled and approximate it for the remainder voxels which are less reliably labeled. For that, we have introduced the membership certainty of voxel labeling to select the voxels which will be considered as control points for Lagrangian interpolation. The reliably labeled voxels allow computing a confident estimation of the bias field. Such enhancements were possible by introducing EM clustering by parts using several sub-volumes that allow to produce usable labeling even if the levels of the different artifacts are high in the MRI volume, in particular the INU level. Experimental results using several MRIs by considering different combinations of noise and INU levels and the comparison with other methods from the literature, showed the efficiency of the proposed method to estimate and correct the bias field in MR images. As most of methods for bias field correction in MRIs, our method does not consider ground-truth MRIs provided in different MRI databases. So, in future work, it is possible to consider other models for hyper-surfaces by integrating data and using machine learning methods. Furthermore, selecting reliable voxels for surface fitting was *ad hoc*, so a machine learning-based selection will enhance the accuracy of the computed hyper-surface that models the bias field.

REFERENCES

[1] E. D. Angelini, T. Song, B. D. Mensh and A. F. Laine, "Brain MRI Segmentation with Multiphase Minimal Partitioning: A Comparative Study," Int. J. Biomedical Imaging, vol. 2007, Article ID: 10526, pp. 1-15, 2007.

[2] E. Ardizzone, R. Pirrone, O. Gambino and S. Vitabile, "Illumination Correction on Biomedical Images," Computing and Informatics, vol. 33, no 1, pp. 175-196, 2014.

[3] J. Ashburner and K.J. Friston, "MRI Sensitivity Correction and Tissue Classification", NeuroImage, vol. 7, no. 4, DOI: 10.1016/S1053-8119(18)31539-8, 1998.

[4] J. Ashburner and K. J. Friston, "Unified Segmentation," NeuroImage, vol. 26, pp. 839-851, 2005.

[5] M. A. Balafar, A. R. Ramli, M. I. Saripan and S. Mashohor, "Review of Brain MR Image Segmentation Methods," Artificial Intelligence Review, vol. 33, no. 3, pp. 261-274, 2010.

[6] B. Caldairou, N. Passat, P. A. Habas, C. Studholme and F. Rousseau, "A Non-local Fuzzy Segmentation Method: Application to Brain MRI," Pattern Recognition, vol. 44, no 9, pp. 1916-1927, 2011.

[7] H. Chang, W. Huang, C. Wu, S. Huang, C. Guan, S. Sekar, K. K. Bhakoo and Y. Duan, "A New Variational Method for Bias Correction and Its Applications to Rodent Brain Extraction," IEEE Trans. Med. Imaging, vol. 36, no. 3, pp. 721-733, 2017.

[8] M. Qin and M. Chen, "A Brain MRI Bias Field Correction Method Created in the Gaussian Multi-scale Space," Proc. of the 9th International Conference on Digital Image Processing (ICDIP 2017), vol. 10420, pp. 1-8, Hong Kong, China, 2017.

[9] Z. Y. Chua, W. Zheng, M.W.L. Chee and V. Zagorodnov, "Evaluation of Performance Metrics for Bias Field Correction in MR Brain Images," J. of Magnetic Resonance Imaging, vol. 29, pp. 1271-1279, 2009.

[10] W. Cong, J. Song , K. Luan, H. Liang., L. Wang, X. Ma and J. Li, "A Modified Brain MR Image Segmentation and Bias Field Estimation Model Based on Local and Global Information," Journal of Computational and Mathematical Methods in Medicine, vol. 2016, Article ID: 9871529, pp. 1-13, 2016.

[11] X. Dai, Y. Lei, Y. Liu, T. Wang, L. Ren, W. J. Curran, P. Patel, T. Liu and X Yang, "Intensity Non-uniformity Correction in MR Imaging Using Residual Cycle Generative Adversarial Network," Journal of Physics in Medicine and Biology, vol. 65, no 21, p. 215025, 2020.

[12] I. Despotovic, B. Goossens and W. Philips, "MRI Segmentation of the Human Brain: Challenges, Methods and Applications, J. Comp. Math. Methods in Medicine, vol. 2015, Article ID: 450341, pp. 1-23, 2015.

[13] L. Dora, S. Agrawal, R. Panda and A. Abraham, "State-of-the-art Methods for Brain Tissue Segmentation: A Review," IEEE Reviews in Biomedical Engineering, vol. 10, pp. 235-249, 2017.

[14] E. Fletcher, O. Carmichael and C. DeCarli, "MRI Non-uniformity Correction through Interleaved Bias Estimation and B-spline Deformation with a Template," Proc. of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 106-109, San Diego, CA, USA, 2012.

[15] J. Gispert, S. Reig, J. Pascau, J. J. Vaquero, P. Garcia-Barreno and M. Desco, "Method for Bias Field Correction of Brain T1-weighted Magnetic Images Minimizing Segmentation Error," Human Brain Mapping, vol. 22, pp. 133-44, 2004.

[16] S. Gonzalez-Villa, A. Oliver, S. Valverde, L. Wang, R. Zwiggelaar and X. Llado, "A Review on Brain Structures Segmentation in Magnetic Resonance Imaging," Journal of Artificial Intelligence in Medicine, vol. 73, pp. 45-69, 2016.

[17] R. Guillemaud, "Uniformity Correction with Homomorphic Filtering on Region of Interest," Proc. of the Int. Conf. on Image Processing (ICIP98), (Cat. No.98CB36269), vol. 2, pp. 872-875, Oct. 1998.

[18] N. J. Habeeb, "Performance Enhancement of Medical Image Fusion Based on DWT and Sharpening Wiener Filter," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 07, no. 02, pp. 118-129, June 2021.

[19] S. Kahali, S. Adhikari and J. K. Sing, "On Estimation of Bias Field in MRI Images: Polynomial *vs.* Gaussian Surface Fitting Method," Journal of Chemometrics, vol. 30, no. 10, pp. 602-620, 2016.

[20] C. Li, R. Huang, Z. Ding, C. Gatenby, D. N. Metaxas and J. C. Gore, "A Lvel Set Method for Image Segmentation in the Presence of Intensity Inhomogeneities with Application to MRI," IEEE Trans. Image Processing, vol. 20, no 7, pp. 2007-2016, 2011.

[21] J. Luo, Y. Zhu, P. Clarysse and I. E. Magnin, "Correction of Bias Field in MR Images Using Singularity Function Analysis," IEEE Trans. Med. Imaging, vol. 24, no. 8, pp.1067-1085, 2005.

[22] A. Madabhushi, J. K. Udupa and A. Souza, "Generalized Scale: Theory, Algorithms and Application to Image Inhomogeneity Correction," Comput. Vis. Image Underst., vol. 101, no. 2, pp.100-121, Feb. 2006.

[23] P.K., Mishro, S. Agrawal, R. Panda and A. Abraham, "Novel Fuzzy Clustering-based Bias Field Correction Technique for Brain Magnetic Resonance Images," IET Image Processing, vol. 14, no. 9, pp. 1929-1936, 2020.

[24] J. W. Murakami, C. E. Hayes and E. Weinberger, "Intensity Correction of Phased-array Surface Coil Images," Magnetic Resonance in Medicine, vol. 35, no. 4, pp. 585-590, 1996.

[25] P. A. Narayana and A. Borthakur, "Effect of Radio Frequency Inhomogeneity Correction on the Reproducibility of Intra-cranial Volumes Using MR Image Data," Magnetic Resonance in Medicine, vol. 33, no. 3, pp. 396-400, 1995.

[26] R. Prakash, R. Meena and R. S. S. Kumari, "Spatial Fuzzy C Means and Expectation Maximization Algorithms with Bias Correction for Segmentation of MR Brain Images," Journal of Medical Systems, vol. 41, no. 1, pp. 1-9, 2017.

[27] B. Scherrer, M. Dojat, F. Forbes and C. Garbay, "Agentification of Markov Model-based Segmentation: Application to Magnetic Resonance Brain Scans," Artificial Intelligence in Medicine, vol. 46, no. 1, pp. 81-95, 2009.

[28] X. Shan, X. Gong and A. K. Nandi, "Active Contour Model Based on Local Intensity Fitting Energy for Image Segmentation and Bias Estimation," IEEE Access, vol. 2018, no. 6, pp. 49817–49827, 2018.

[29] J. G. Sled and G. B. Pike, "Standing Wave and RF Penetration Artifacts Caused by an Elliptic Geometry: An Electrodynamics Analysis of MRI," IEEE Trans. Med. Imaging, vol. 17, no. 4, pp. 653-662, 1998.

[30] S. M. Smith, "Fast Robust Automated Brain Extraction," Human Brain Mapping, vol. 17, no. 3, pp. 143-155, 2002.

[31] S. Song, Y. Zheng and Y. He, "A Review of Methods for Bias Correction in Medical Images," Biomedical Engineering Review, vol. 1, no. 1, DOI: 10.18103/bme.v3i1.1550, 2017.

[32] J. Song and Z. Zhang, "Brain Tissue Segmentation and Bias Field Correction of MR Image Based on Spatially Coherent FCM with Nonlocal Constraints," Comput. Math. Methods Medicine, vol. 2019, Article ID: 4762490, pp. 1-13, 2019.

[33] K. R. Sreenivasan, M. Havlicek and G. Deshpande, "Nonparametric Hemodynamic Deconvolution of FMRI Using Homomorphic Filtering," IEEE-Trans. Med. Imaging, vol. 34, no. 5, pp. 1155-1163, 2015.

[34] M. Styner, C. Brechbuhler, G. Szekely and G. Gerig, "Parametric Estimate of Intensity Inhomogeneities Applied to MRI," IEEE Trans. Med. Imaging, vol. 19, no. 3, pp. 153-165, 2000.

[35] P. Vemuri, E. G. Kholmovski, D. L. Parker and B. E. Chapman, "Coil Sensitivity Estimation for Optimal SNR Reconstruction and Intensity Inhomogeneity Correction in Phased Array MR Imaging," Proc. of the 19th International Conference on Information Processing in Medical Imaging (IPMI 2005), pp. 603-614, Glenwood Springs, USA, July 10-15, 2005.

[36] U. Vovk, F. Pernus and B. Likar, "A Review of Methods for Correction of Intensity Inhomogeneity in MRI," IEEE Transactions on Medical Imaging, vol. 26, no. 3, pp. 405-421, March 2007.

[37] L. Wang, J. Zhu, M. Sheng, A. Cribb, S. Zhu and J. Pu, "Simultaneous Segmentation and Bias Field Estimation Using Local Fitted Images," Pattern Recognition, vol. 74, pp. 145-155, 2018.

[38] A. I. Zayed and P. L. Butzer, "Lagrange Interpolation and Sampling Theorems," Non-uniform Sampling, pp. 123-168, DOI: 10.1007/978-1-4615-1229-5_3, Springer, 2001.

"A Hyper-surface-based Modeling and Correction of Bias Field in MR Images", D. Azzouz and S. Mazouzi.

ملخص البحث:

إنّ التّعامل مع العمليّات الاصطناعيّة في الصّور الطبّيّة أمر ضروري لأداء العديد من المهامّ، بما في ذلك التّجزئة. نقدّم في هذه الورقة طريقة مبتكرة لتصحيح حقل الانحياز في التّصوير بالرّنين المغناطيسي. فباستخدام نتائج التّجزئة التي يتمّ الحصول عليها عن طريق العنقدة المعدّلة المتعلّقة بتعظيم التّوقّع، تتمّ ملاءمة حقل الانحياز كسطح فوقيّ في حيّز كثير السّطوح رباعيّ الأبعاد.

بعد ذلك، يجري تصحيح حقل الانحياز بناءً على حقيقة أنّ النّقط المجسّمة التي تتبع للنّسيج نفسه يجب أن تكون لها شدّة الإضاءة نفسها في الصّورة كلّها. وهكذا، فبعد عملية سريعة وغير دقيقة لوسم النّقط المجسّمة عن طريق العنقدة تبعاً للأجزاء، يتم حساب حقل الانحياز للنّقط المجسّمة الموسومة بشكل موثوق. أمّا بالنسبة للنّقط المجسّمة الموسومة على نحوِ أقلّ موثوقية، فإنّ حقل الانحياز يتم استيفاؤه باستخدام سطح فوقيّ يجري تقديره بواسطة استيفاء "لاجرانج" رباعي الأبعاد.

لقد قمنا بتقييم الطريقة المقترحة عن طريق مقارنة نتائج التّجزئة بوجود تصحيح حقل الانحياز وغياب تصحيحه. كذلك استخدمنا معامل التّغيّر ضمن حجم صورة الرّنين المغناطيسي. وكانت نتائج التقييم أفضل من حيث التّجزئة ومعامل التّغيّر – وبشكل ملموس- بعد تصحيح حقل الانحياز باستخدام الطريقة المقترحة.

INTRODUCING A NEW ROUTING ALGORITHM FOR WIRELESS NETWORKS ON CHIP USING REINFORCEMENT LEARNING

Zohreh Harati[1], Esmaeel Tahanian[2], Alireza Tajary[3] and Mansoor Fateh[4]

(Received: 16-Apr.-2021, Revised: 13-Jun.-2021, Accepted: 3-Jul.-2021)

ABSTRACT

Wireless network on chip (WNoC) can be used as an alternative to bus technology in high-core chips in which the multi-hop paths between far apart cores are replaced with a wireless single-hop link. The main reason for using wireless communication is to reduce latency as well as power consumption. According to the limitation of resources, the performance of the WNoC is sensitive to the routing algorithm. While an appropriate routing algorithm reduces latency, it should avoid deadlock. In this paper, we propose a novel routing algorithm using Q-learning, which is one of the reinforcement learning methods for balancing wireless network traffic on the chip. Using such an algorithm, the nodes can make decisions based on congestion conditions in the network when transferring flits from the source node to the destination one. The simulation results show that using the proposed reinforcement learning for routing the packets considerably improves the performance of the network; more precisely, the system performance is improved by 8% compared with the previous related works.

KEYWORDS

Wireless network on chip (WNoC), Q-learning algorithm, Reinforcement learning (RL), Routing algorithm, Deadlock.

1. INTRODUCTION

Coincidence with the development of the electronic world and the increase in the ability to design circuits at the nanoscale, integration in the design of electronic devices has become the talk of the day and efforts to design and build integrated devices continue. These efforts have created a new field called the System on Chip (SoC). The SoC tries to integrate the processing, communication and interface cores. One of the problems with this integration is how to connect and communicate between these cores.

NoC [1]-[2] is a new solution for interconnection within the SoC, which was introduced in 1999. In traditional solutions, intermediate connections were made using the structure of the bus. As circuits became more compact, bus-based solutions lost their effectiveness in contrast to the need for new technologies. Crossings began to restrict and, at worst, block traffic. In networks on chip, there are networks like computer networks, in which different parts communicate with each other through this network by sending packaged data. An NoC like a computer network includes cores, routers that route traffic between cores and wires that connect devices to routers and routers to each other.

In this network, packets or flits are used to send data. In general, the package consists of at least three parts, which include the header, body and end of the package tail. Each package has only one header, one or more bodies and one end of the package tail. When sending data, the packet header is sent first and the path is reserved for data transmission. When one or more bodies are sent, by sending the end of the package tail, the reserved path is released so that it can be reserved by other nodes. In a WNoC [33]-[34], multi-hop wired paths between far apart cores are replaced by high-bandwidth single-hop long-range wireless links. Consequently, reduction of average hop count leads to better performance of the network and especially reduces latency and power consumption. The idea of the NoC using wireless links has been proposed for the first time in 2010 [3] and has so far been considered from different aspects such as routing algorithms. The routing algorithm identifies the path between the source router and the destination router. Packets have two strategies for arriving at the destination in WNoC. In the first strategy, the packets use the nearest wireless nodes to the sender and receiver for arriving at the destination. In the second strategy, the packets are routed only through wired links. Clearly, the path

Z. Harati, E. Tahanian, A. Tajary and M. Fateh are with Department of Artificial Intelligence Engineering, Shahrood University of Technology, Shahrood, Iran. Emails: [1]zohreh.harati@gmail.com, [2]e.tahanian@shahroodut.ac.ir, [3]tajary@shahroodut.ac.ir and [4]mansoor_fateh@shahroodut.ac.ir

with a minimum number of hops would be chosen by the traveling packets as the shortest path. So far, the most important routing methods that have been proposed include source and distributed routing [4], [5], [6], [7], deterministic and adaptive routing [8]-[10], minimal and non-minimal routing [11]-[14], thermal-aware routing [15]-[18] and fault-tolerant routing [19]-[22], [37].

One of the most important routing algorithms is the learning-based algorithm that can be categorized as an adaptive routing algorithm [46]. In the deterministic routing algorithm, it is decided only based on the source and destination addresses and all packets that have the same source and destination addresses pass the same route. But in the adaptive routing algorithm, in addition to the source and destination addresses, the network time traffic is also effective in determining the route. Therefore, packets that have the same source and destination addresses may use different paths with different delays for data transmission.

Reinforcement learning is a type of learning in which the correct action in each situation is determined by a standard. It is the task of the teaching agent to learn the best action in any situation by having information [38]-[40]. This is part of the specific strengths of reinforcement learning. With the help of reinforcement learning, often the complexities of the decision can be solved with the least amount of information required.

In reinforcement learning, the main purpose of learning is to perform a task or achieve a goal without which the learning agent is fed with external direct information. In reinforcement learning, when the agent performs a task that makes him closer to his goal, rewards are received and the goal is to take steps to maximize the agent’s reward for the long term [23]-[24]. In reinforcement learning, rewards and punishments are used as signals to improve the final performance of the system [10], [25].

In this paper, we propose a routing algorithm based on reinforcement learning which can execute on the wireless chip and uses flit to move data in a network on a chip. In the proposed method, we use a reinforcement learning method called Q-learning, which is a value-based method. This learning has a function Q the inputs of which are states and actions. To learn this function uses a table in which rows are states and columns are actions. If we have an agent who is at first home and can perform certain movements to act, to continue the activity of the agent, he must look at the table to obtain the value of Q from the starting position and based on specific actions and choose whichever has a higher Q value, receive the reward and update the value of Q based on the Bellman relation. The function Q indicates how much the agent may be rewarded on its path.

Q-routing is a network routing method based on the Q-learning algorithm. Each node makes its own routing decisions based on information about its neighboring nodes. The node stores tables of values Q that estimate the quality of the alternative paths. These values are updated each time a node sends a packet to one of the neighbors. Thus, with packets of node paths, the values Q gradually hold more information. This routing algorithm can adapt to the network. In this way, if the packet path from the source node to the destination node is crowded, it changes the route and chooses the less crowded route. Redirection by the algorithm is based on the information in the tables of its nodes, where the information of these tables is updated based on the information obtained in each node movement. In fact, learning in this routing algorithm is the product of changing table values.

The remainder of this paper is organized as follows. In Section 2, the related work is discussed. XY routing algorithm, wireless routing algorithm and flit and Q-learning algorithm are given in Section 3. In Section 4, the proposed method for NoCs is explained. The results are reported in Section 5, while the summary and conclusion are given in the last section.

2. RELATED WORK

In the past, a lot of research has been done to improve the performance of NoC routing. Recently, learning-based algorithms have been presented that can be categorized as adaptive routing algorithms. In the deterministic routing algorithm, the paths of the packets are determined only based on the source and destination addresses and all the packets that have the same source and destination addresses pass the same route. But in the adaptive routing algorithm, in addition to the source and destination addresses, the network time traffic is also effective in determining the route. Therefore, packets that have the same source and destination addresses may use different paths with different delays for data transmission.

Farahnakian et al. used reinforcement learning to balance traffic in the network. In this work, the Q - learning technique is used and the state is the node, the agent is the packet and the action is the output port selection. In this way, there is a table for each node that shows the different states of movement of the packet from that node in the network and the values of this table are equivalent to the delay that will be spent to reach the packet from the source node to destination node. When a packet reaches a node, it checks the table, because the goal is to select the least crowded path. If the delay value is assumed to be positive, it selects the lowest value and sends the packet in that direction. When the packet is sent through this node and the receiving node receives it, the receiving node sends its information about the status of its ports to the sender node through the learning packet and the sending node uses this information to update its table. This routing is a one-way routing, because when a packet is sent from node x to node y, only the sender node table, node x, is updated. Considering the packet instead of the flit is the main drawback of this work. In addition, this method was presented only for wired NoCs. Figure 1 shows an example of one-way routing.

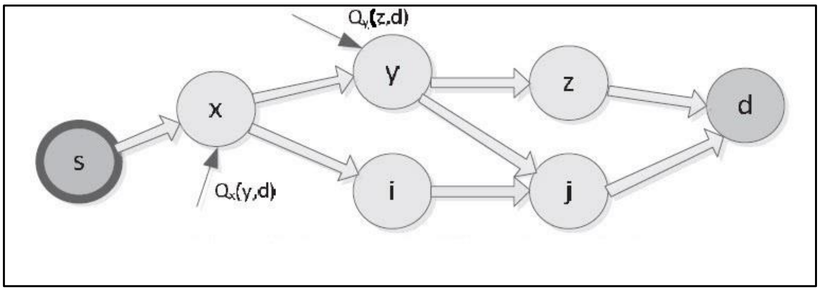


Figure 1. An example of one-way routing [26].

In reference [27] to solve the traffic problem in networks on chip, two-way reinforcement learning has been used. In this method, after selecting the path, the sender node also sends information about the congestion status of its ports when sending the packet. This information is used to update the receiver node learning table. When the packet is sent with this added information, the receiving node receives this information and extracts the required information from the received packet and sends its related information to the sending node with the learning packet. With this information that the receiver and the sender node receive, they update their tables. This is why it is said two-way. The proposed method in this reference finds the optimal path faster and improves the performance of the network. Again, this reference has considered only wired NoC as well as packet instead of the flit. Figure 2 shows an example of two-way routing.

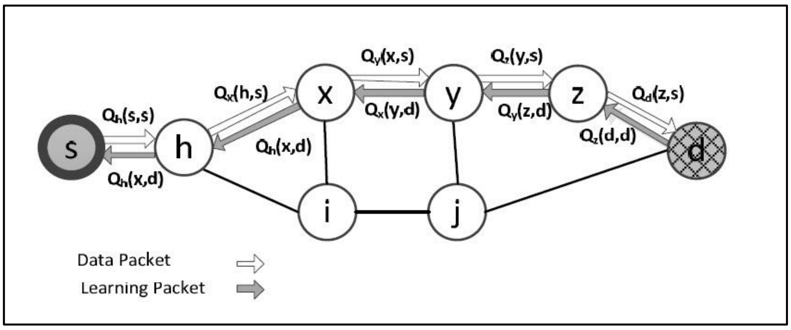


Figure 2. An example of two-way routing [27].

In reference [28], the performance of the network on the chip is optimized using reinforcement learning. Farahnakian et al. in [29] proposed an adaptive routing algorithm that distributes traffic using a learning method across the network. In reference [30], a congestion-aware routing algorithm based on Q-learning is presented, which divides the network into several clusters and each cluster maintains a table. This table stores local and general congestion information about alternative routes to send packets to the destination cluster. Each cluster can select a low-density output channel based on table information. Also, to further update the tables, both learning packages and data participate in the publication of congestion information. In this reference, flit is used for data transmission, but wireless links are not used.

It is worth noting that in all mentioned routing algorithms based on reinforcement learning, the data was transferred in a wired network and a packet was used to transfer data. In this paper, we apply reinforcement learning to improve the routing in a wireless network on chip when the flits are used to transfer data.

3. BACKGROUND

This section briefly reviews XY routing algorithm, wireless routing algorithm and Q-learning algorithm.

3.1 XY Routing Algorithm

By using the XY routing algorithm which is a kind of distributed deterministic routing algorithms, the flits first follow the X-path, then they move in the Y direction to reach their destination.

3.2 Wireless Routing Algorithm

Using wireless connections, the algorithm sends the packet faster from the source node to the destination one [41]-[43], [45]. In this kind of algorithms, when a packet reaches a node, two cases are checked:

- Send packet from the source to the destination by XY method.
- Send a packet from the source to the destination by sending the packet to the nearest wireless router to the source, receiving it at the nearest wireless router to the destination and then sending the packet to the destination node.

The selection of each of the above-mentioned cases is done by calculating the Cartesian distance between the source and the destination, once without considering the wireless node and once using the wireless node. If the first distance is less than the second distance, the XY routing algorithm method will be used; otherwise, the wireless XY routing algorithm is selected.

If there is no wireless node, the Cartesian distance (CD) is equal to:

CD = |The distance between the source column and the destination column| + | distance between source row and destination row| (1)

and if there is a wireless node, the Cartesian distance is equal to [41]:

CD = |Cartesian distance between the destination and the nearest wireless node to the destination |+ |Cartesian distance between the source and the nearest wireless node to the source | + the cost of using wireless node. (2)

3.3 Q-learning Algorithm

Q-learning is a reinforcement learning technique in which by learning a state-action function, the agent follows a specific policy for performing different movements in different situations [44]. In the Q-learning algorithm, each state-action pair is assigned a value of Q (s, a), which is the sum of the rewards received. When the agent starts from states, operates a and follows the existing policy, until it converges to the optimal value, Equation (3) (known as Bellman's relation) is used to update.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t(s_t, a_t) \times [R(s_t) + \gamma \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{3}$$

The core of the algorithm consists of a simple iterative update. In this way, the previous values are modified based on the new information. The Q-learning algorithm uses tables to store data. Each table maintains four fields named Current-State Space, Next-State Space, Action Space and Q-value. Current-State Space refers to everything that the agent currently perceives, while Next-State Space is determined by the Current-State and the actions selected by the agents. The Action Space indicates the actions that the agent can perform. Each Q-value is accessible by Q(s, a), representing the expected reinforcement of taking action in states. Algorithm 1 shows this procedure.

Algorithm 1. Q-learning algorithm.

```
Input: PIR: packet Injection Rate, T: simulation Time, WN: wireless Nodes Set, lr: learning rate, $m \times n$: network size, df: discount factor.
Output: Final Q $(s_t, a_t)$
ST ← 1,2,3, …, T; N ← 1,2, …, $m \times n$
Initialize Q(s,a) arbitarily
For  i ∈ ST  do
    Initialize s
    For  j ∈ N do
        Select action of agent from s using policy derived from Q(e.g, €-greedy)
        $\mathbf{Q}(s_t, a_t) \leftarrow \mathbf{Q}(s_{t+1}, a_{t+1}) + a[r_{t+1} + \gamma max_a * \mathbf{Q}(s_{t+1}, a_{t+1}) - \mathbf{Q}(s_t, a_t)]$
        s ←s';
    end
end
Return Q$(s_t, a_t)$
```

4. The Proposed Method

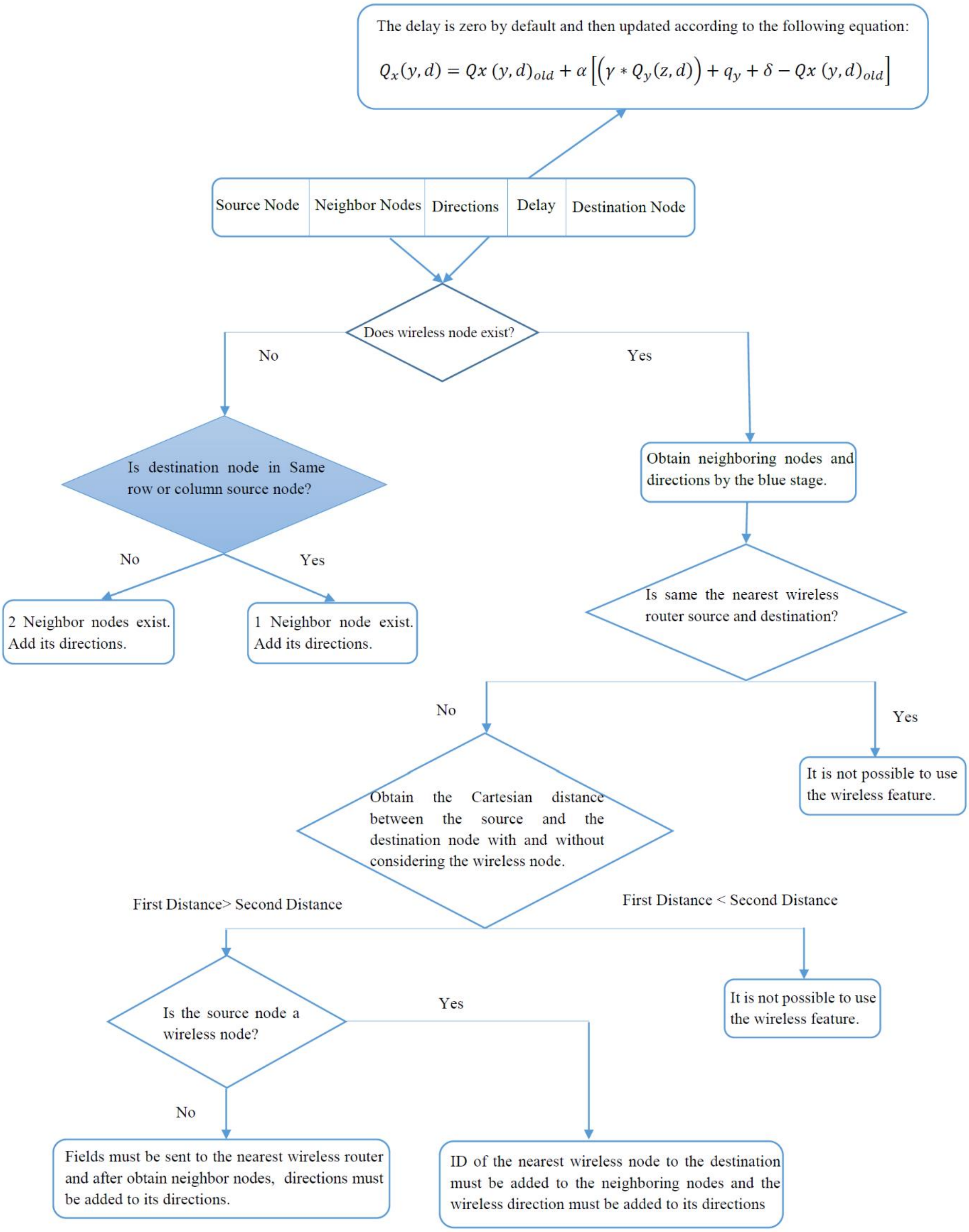
In this section, to use Q-learning in wireless NoC, we first properly modify the conventional Q-table. Then, we explain how to fill this table. Finally, we describe the proposed Q-learning algorithm for the wireless NoC.

4.1 The Customized Q-table for the Wireless NoC

The customized Q-table for node 1 in a 3*3 WNoC is illustrated in Table 1. The table contains one field for the source, one field for the destination, four fields for the neighboring nodes (indicating the nodes we must first go to when moving from the source to the destination), four fields for the direction and four fields for the delay.

Table 1. Table values in a 3 * 3 network with two nodes; namely, 1 and 6, as the wireless nodes.

Source	Neighboring node				Direction				Delay				Destination
Node1	0				3				0				Node0
Node1	1				4				0				Node1
Node1	2				1				0				Node2
Node1	0	4	6		3	2	5		0	0	0		Node3
Node1	4				2				0				Node4
Node1	2	4			1	2			0	0			Node5
Node1	0	4	6		3	2	5		0	0	0		Node 6
Node1	4		6		2		5		0		0		Node 7
Node1	2	4	6		1	2	5		0	0	0		Node 8


The delay is zero by default and then updated according to the following equation:
$Q_x(y,d) = Qx\,(y,d)_{old} + \alpha\left[\left(\gamma * Q_y(z,d)\right) + q_y + \delta - Qx\,(y,d)_{old}\right]$
Source Node
Neighbor Nodes
Directions
Delay
Destination Node
Does wireless node exist?
No
Yes
Is destination node in Same row or column source node?
Obtain neighboring nodes and directions by the blue stage.
No
Yes
2 Neighbor nodes exist. Add its directions.
1 Neighbor node exist. Add its directions.
Is same the nearest wireless router source and destination?
No
Yes
It is not possible to use the wireless feature.
Obtain the Cartesian distance between the source and the destination node with and without considering the wireless node.
First Distance> Second Distance
First Distance < Second Distance
Is the source node a wireless node?
Yes
It is not possible to use the wireless feature.
No
Fields must be sent to the nearest wireless router and after obtain neighbor nodes, directions must be added to its directions.
ID of the nearest wireless node to the destination must be added to the neighboring nodes and the wireless direction must be added to its directions

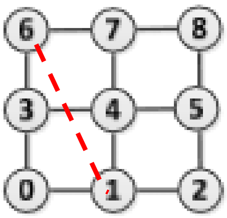


Figure 3. A typical 3*3 mesh network with a wireless link between nodes 1 and 6.

- In that table, there is one row for all destinations. For example, in a 3 * 3 mesh, we have 9 rows for each source node, which is from 0 to 8.
- Each row contains information about sending data from the source node to the destination node.
- There is a table for each node in the network.

Filling the four fields of the neighbor nodes is as follows:

- If there is no wireless node in the network and the destination node is in the same row or column of the source node, the source node has one direction to send data to the destination node and therefore there is only a neighboring node in the table. On the other hand, if the source node and the destination node are not in the same row or column, there are two directions for sending data. Consequently, the source node has two neighboring nodes.
- If there is a wireless node in the network, in addition to obtaining the neighboring nodes, the use of wireless capability should be checked.

If the wireless transmission is possible, two cases may occur: first, the source node is a wireless node and the other source node is not a wireless node. If the source node is wireless, the ID of the nearest wireless node to the destination must be added to the neighboring nodes. This is because the next node is the wireless node to which the fields must be sent and the wireless direction must be added to its directions. If the source is not wireless, the fields must be sent to the nearest wireless router to use the wireless capability.

If the nearest wireless source and destination are not the same, we should consider whether it is good to use wireless capability. To do this, we must obtain the Cartesian distance between the source and the destination node with and without considering the wireless node. If the second distance is less than the first one, the wireless capability will not be used; otherwise, the wireless capability will be used. Furthermore, the delay values when creating the table for each node are zero by default.

To calculate and update the delay values, Equation (4) is used, which simulates the Bellman relation [35] in the NoC for data transmission [26].

$$Q_x(y,d) = Qx\ (y,d)_{old} + \alpha\left[\left(\gamma * Q_y(z,d)\right) + q_y + \delta - Qx\ (y,d)_{old}\right] \tag{4}$$

This relation calculates the amount of new latency when sending a packet from node *x* to node *d* through the neighboring node y. In this regard, α is the learning rate that determines the rate of newer information. In addition, *qy* indicates how long the packet sent from node *x* waits in the input buffer of node *y*. δ indicates the length of time that a packet is sent from node *x* to node y. Moreover, $Q_y(z,d)$ indicates the length of time that the packet goes from node *y* to node *d* with the help of the neighboring node *z*. $Qx\ (y,d)_{old}$ indicates the amount of delay in the pre-update table and γ is the discount factor that determines the importance of future rewards and therefore affects the value of $Q_y(z,d)$.

Each of the above values is obtained as follows: The value $Qx\ (y,d)_{old}$ exists in the table. To obtain the value of $Q_y(z,d)$, a separate table must be created with node tables in each cell and this table must be publicly available. Of course, this only allows access to the next node table. Therefore, it must be checked which row of the destination equals the intended destination. When the desired row is obtained, the delay the value of which is the least and is opposite to -1 is selected as the delay from node y to destination d.

To calculate δ and q_y, when a packet is placed in one direction, the sc_time_stamp () function is used to find the packet's time in the desired direction and when the packet is received and a confirmation message is issued, its time will be counted. The difference between the two times indicates the amount of delay.

4.2 The Proposed Routing Algorithm

After creating the table and explaining how to update it, the performance of the proposed algorithm is examined. This algorithm considers the amount of delay and which of the delay values is less; its direction is considered as the next direction of packet movement.

In general, the procedure of the program is that for each node, a table is created and completed using the given information and for the delay fields, zero values are placed first. Then, using the proposed routing algorithm, a packet path is selected. Because initially the amount of delay is considered zero by fefault, the direction of the first neighbor node is chosen to move the packet. After selecting the path, the delay value is calculated and updated using Equation (4) and the data will continue to move on the path of selection and this procedure continues to reach the destination.

5. Experimental Result

Noxim simulator [31] has been used to implement the learning-based routing algorithm and to perform its simulation. Noxim has a command line that can give several parameters as input to the simulator, such as buffer size, type of routing algorithm, location of wireless nodes, number of virtual channels, simulated duration and traffic type, …etc. The simulation output, including power, packet transmission delay, number of injected fields, the percentage of packets received *via* wireless, …etc., is given to the user.

In these experiments, the performance of the proposed method is compared with the XY and the WirelessXY routing algorithm. These simulations are conducted on a 4 * 4 mesh with two wireless links; namely, nodes 1 & 6 and the performance of routing algorithms is evaluated based on latency curves. It is assumed that data packets and learning packets have different lengths. The simulation time is set to 5000 ns and the buffer size is set to 4. We have also used three different traffic patterns, Transpose, Hotspot and Random, to display and compare results [36]. For the Hotspot case, 5% of the generated traffic by all cores have the same destination which can be chosen randomly. The destination of the other 95% of the generated packets by a core will be chosen randomly. Figure 4 shows the average latency as a function of the average data injection rate in random traffic.

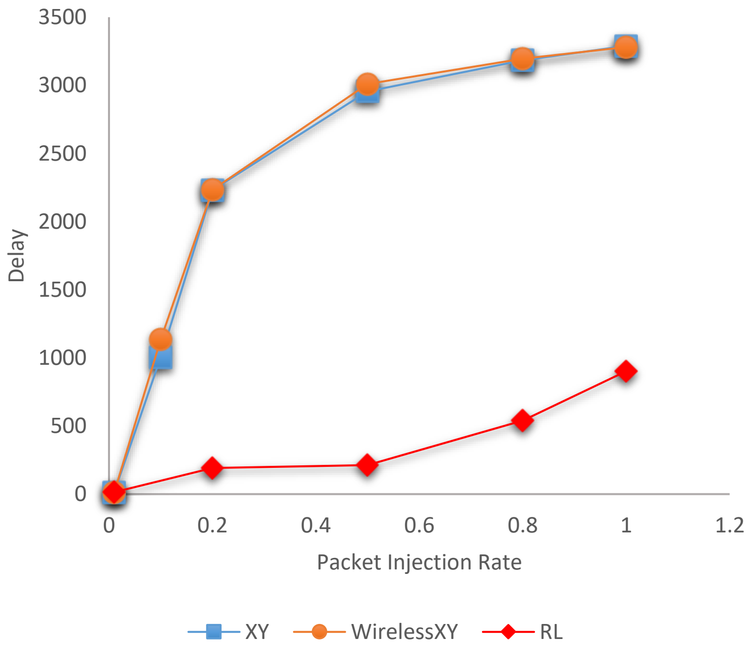


Figure 4. Comparison of delay for the proposed algorithm, XY and wirelessXY in random traffic.

The horizontal axis of the diagram shows how likely it is to be injected per clock for each core and the higher the injection rate, the higher the network load. In random traffic [32], each node with a random probability sends a packet to another node. The destination of different packets is determined using a uniform distribution randomly .As can be seen from the results, the proposed algorithm has resulted in less latency than other algorithms and in this case, the proposed algorithm has improved latency by at least up to 9%.

Figure 5 shows the average latency as a function of the average data injection rate in transpose traffic. In transpose traffic, a node (j, i) can only send packets to one node (i, j). As can be seen from the results, in this type of traffic, the proposed algorithm has led to less latency than other algorithms and in this case, the proposed algorithm has improved latency by at least 8%.

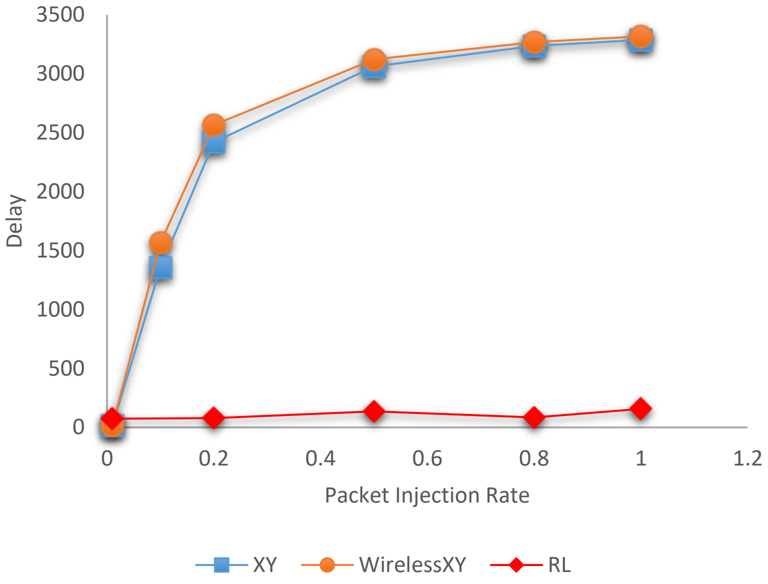


Figure 5. Comparison of delay for the proposed algorithm, XY and wirelessXY in transpose traffic.

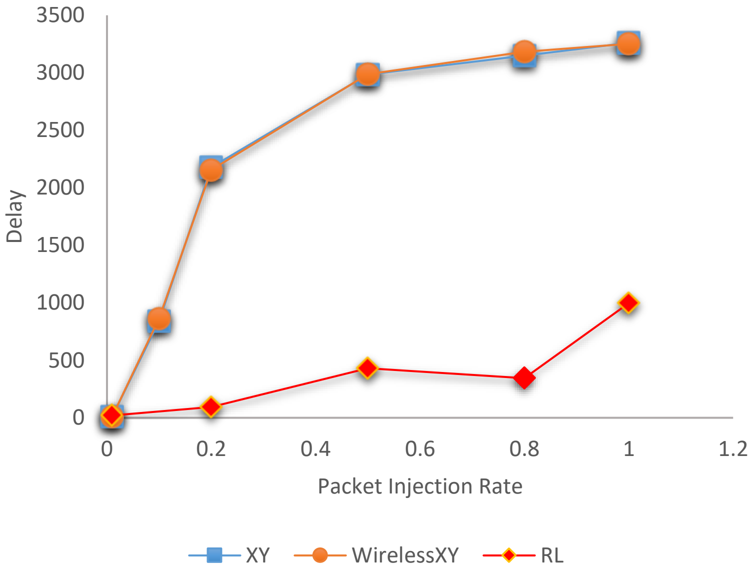


Figure 6. Comparison of delay for the proposed algorithm, XY and wirelessXY in hotspot traffic.

Figure 6 shows the average latency as a function of the average data injection rate in hotspot traffic. In hotspot traffic, there are one or more nodes selected as points that receive more traffic in addition to regular monotonous traffic. For this case, the proposed algorithm has led to less latency than other algorithms and in this case, the proposed algorithm has improved latency by at least 12%.

The results of the proposed algorithm in all cases in comparison with the WirelessXY and XY algorithms is better, since the proposed algorithm is adaptable to the congestion conditions; that is, if the packet encounters a crowded route on the way to the destination, it changes its route and does not wait for the route to be released, which causes it to be less delayed than in other algorithms.

To illustrate the movement of a packet in the path source node to the destination node, the following curves in Figure 7 are shown. These diagrams show the values of delay for the middle nodes in a 3*3 mesh where the source node is node zero and the destination node is node 5.

In this figure, the middle nodes; namely, node 1 and node 3 are examined. In addition, for each time interval, one graph is straight and in the same interval, the other graph is sloping. The sloping diagram shows that the packet is moving in this direction and the straight diagram shows that the moving packet does not exist in this path. Packet movement in both paths is performed according to the proposed algorithm by examining the values of delay and how it works, considering that the values of delay are zero by default. The direction of the first neighbor node is selected as the direction of packet movement.

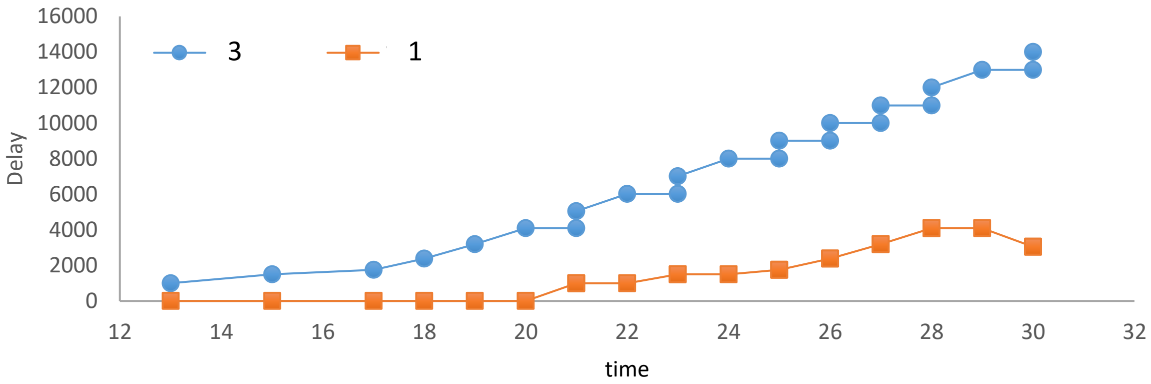


Figure 7. Graph of how the packets move at a learning rate of 0.5 for nodes 1 and 3.

Therefore, the packet starts moving in the path of node 3 for up to 20 ns. At this time, the proposed algorithm, after examining the amount of delay in the two nodes, realizes that the amount of latency in node 1 is less than in node 3, so it puts the direction of movement of the package in the path of node one. The delay is checked again in 21 ns and this time, the source node has placed the packet in the path of node 3 and the process of selecting a less crowded path continues. Finally, Figures 8 and 9 compare the throughput and power consumption of the WNoC for different routing algorithms; namely, RL-based, XY and WirelessXY.

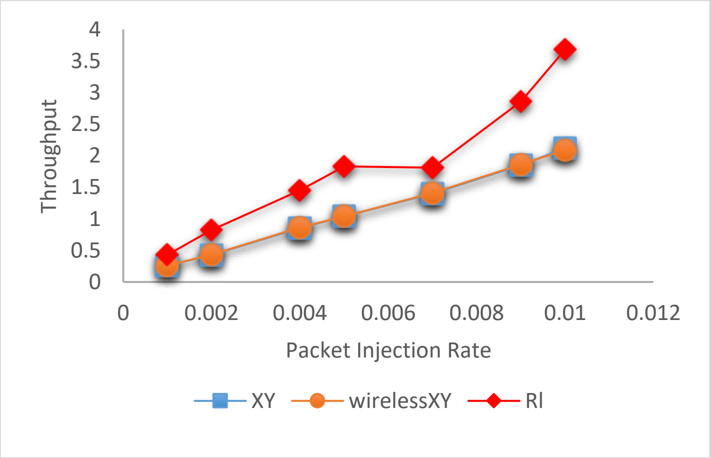


Figure 8. Comparison of throughput for the proposed algorithm, XY and WirelessXY in random traffic.

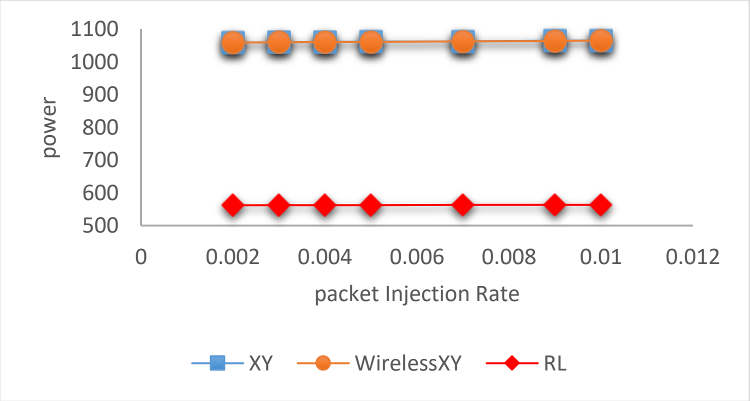


Figure 9. Comparison of power consumption for the proposed algorithm, XY and WirelessXY in random traffic.

5.1 Comparison with Previous Works

Figures 10 and 11 compare the proposed algorithm with reference [30]. The horizontal axis shows the injection rate of the package and the vertical axis shows the delay or time it takes the package to reach the destination. In these figures, a network with dimensions of 8*8 for two different traffics; namely, random traffic and hotspot is considered.

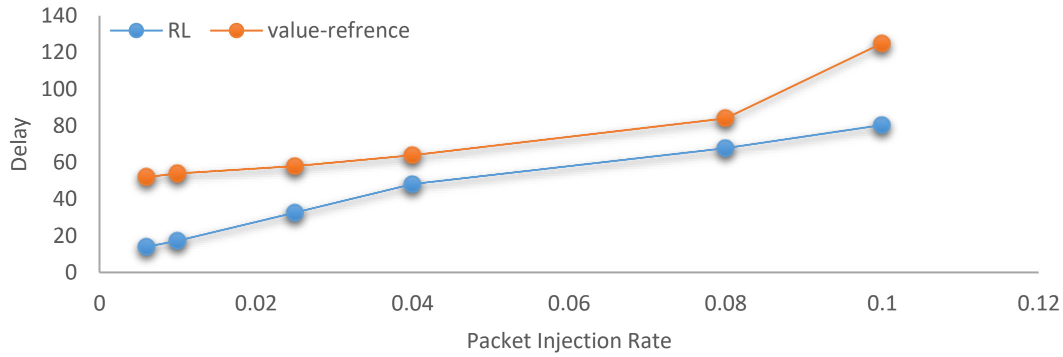


Figure 10. Comparison of delay for the proposed algorithm and the algorithm proposed in [30] for random traffic in an 8*8 network.

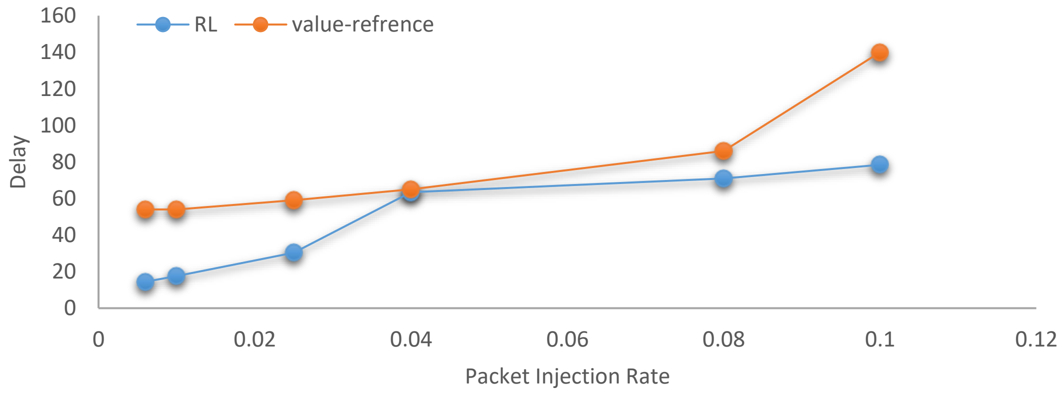


Figure 11. Comparison of delay for the proposed algorithm and the algorithm proposed in [30] for hotspot traffic in an 8*8 network.

As shown in Figures 10 and 11 for an 8*8 NoC, the delay evaluated using the proposed algorithm is always less for both traffic types.

The better results in the proposed algorithm are due to the use of wireless links. These links lead to a tendency for nodes to send their data (for ease of data transmission) *via* these links and this could significantly reduce congestion on the network and allow less delay to the destination.

6. Conclusion

In this paper, a routing algorithm based on reinforcement learning in a wireless chip network is proposed, which can make decisions based on congestion conditions in the network when transferring flits from the source node to the destination one. This algorithm uses the Q-learning method. In this method, there is a table for each node that shows the different states of packet movement of that node in the network and the values of this table are equivalent to the delay which is spent to get the packet from the source node to the destination node. This table is first filled with the initial value of zero and then updated by moving from one node to another node. When a packet comes to a node, the table of the node is considered. If the delay value is assumed to be positive, it will select the delay's lowest value and send the packet in that direction. In fact, learning in this algorithm is done by updating and changing the table values of each node. To evaluate the algorithm, this method was compared with the XY and the WirelessXY algorithms in different types of traffic. The experimental results show that the reinforcement learning-based routing algorithm can improve the delay at least by around 8% for all traffic types.

The reinforcement learning algorithm has some disadvantages and limitations as compared to other learning algorithms, as in large networks on the chip where each node contains many data, the reinforcement learning algorithm cannot analyze these networks. Therefore, as future work, we intend to use deep reinforcement learning rather than reinforcement learning to develop our approach to develop more complex environments.

REFERENCES

[1] S. Kundu and S. Chattopadhyay, Network-on-Chip: The Next Generation of System-on-Chip Integration, Taylor & Francis, 2014.

[2] R. Venugopalan, S. Kumar Goel and Y.-H. Lee, "Network-on-Chip System and a Method of Generating the Same," U.S. Patent Application 16/879,567, Filed September 3, 2020.

[3] A. Ganguly, K. Chang, S. Deb, P. Pratim Pande, B. Belzer and C. Teuscher, "Scalable Hybrid Wireless Network-on-Chip Architectures for Multicore Systems," IEEE Transactions on Computers, vol. 60, no. 10, pp. 1485-1502, 2010.

[4] J. Flich, S. Rodrigo and J. Duato, "An Efficient Implementation of Distributed Routing Algorithms for NoCs," Proc. of the 2nd ACM/IEEE Int. Symposium on Networks-on-Chip (NOCs 2008), pp. 87-96, 2008.

[5] Z. Mogharrabi-Rad and E. Yaghoubi, "ADFT: An Adaptive, Distributed, Fault-tolerant Routing Algorithm for 3D Mesh-based Networks-on-Chip," International Journal of Internet Technology and Secured Transactions, vol. 10, no. 4, pp. 481-490, 2020.

[6] R. Bishnoi, "Hybrid Fault Tolerant Routing Algorithm in NoC," Perspectives in Science, vol. 8, pp. 586-588, 2016.

[7] S. Mubeen and S. Kumar, "Designing Efficient Source Routing for Mesh Topology Network on Chip Platforms," Proc. of the 13th IEEE Euromicro Conference on Digital System Design: Architectures, Methods and Tools, pp. 181-188, 2010.

[8] Z.-S. Chen, Y. Zhang, Z. Peng and J.-H. Jiang, "A Deterministic-path Routing Algorithm for Tolerating Many Faults on Wafer-level NoC," In IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1337-1342, Florence, Italy, 2019.

[9] T. A. Eltaras, W. Fornaciari and D. Zoni. "Partial Packet Forwarding to Improve Performance in Fully Adaptive Routing for Cache-coherent NoCs," Proc. of the 27th IEEE Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), pp. 33-40, Pavia, Italy, 2019.

[10] M. M. Rahaman, P. Ghosal and T. Subhra Das, "Latency, Throughput and Power Aware Adaptive NoC Routing on Orthogonal Convex Faulty Region," Journal of Circuits, Systems and Computers, vol. 28, no. 04, DOI: 10.1142/S0218126619500555, 2019.

[11] M. Sun, Q. Liu, B. Yan and X. Wang, "Minimally Buffered Router and Deflection Routing Algorithm for 3D Mesh NoC," Proc. of Recent Developments in Intell. Computing, Communication and Devices, Part of the Advances in Intelligent Systems and Computing Book Series, vol. 752, pp. 515-522, 2019.

[12] M. Schoeberl, L. Pezzarossa and J. Sparsø, "A Minimal Network Interface for a Simple Network-on-Chip," Proc. of the International Conference on Architecture of Computing Systems (ARCS 2019), Part of the Lecture Notes in Computer Science Book Series, vol. 11479, pp. 295-307, 2019.

[13] Song, Yang and Bill Lin. "Uniform Minimal First: Latency Reduction in Throughput-optimal Oblivious Routing for Mesh-based Networks-on-Chip." IEEE Embedded Systems Letters, vol. 11, no. 3, pp. 81-84, 2019.

[14] L. Wang, X. Wang, H.-F. Leung and T. Mak, "A Non-minimal Routing Algorithm for Aging Mitigation in 2D-mesh NoCs," IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, vol. 38, no. 7, pp. 1373-1377, 2018.

[15] K.-C. Chen, "Game-based Thermal Delay-aware Adaptive Routing (GTDAR) for Temperature-aware 3D Network-on-Chip Systems," IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 9, pp. 2018-2032, 2018.

[16] W. Zhang and Y. Ye, "A Table-free Approximate Q-learning Based Thermal-aware Adaptive Routing for Optical NoCs," IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, vol. 40, no. 1, pp. 199-203, 2020.

[17] Y. Ye, W. Zhang and W. Liu, "Thermal-aware Design and Simulation Approach for Optical NoCs," IEEE Trans. on Computer-aided Design of Integrated Circuits and Sys., vol. 39, no. 10, pp. 2384 – 2395, 2019.

[18] N. Shahabinejad and H. Beitollahi, "Q-thermal: A Q-learning Based Thermal-aware Routing Algorithm for 3D Network On-Chips," IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 10, no. 9, pp. 1482 – 1490, 2020.

[19] T. H. Vu, O. M. Ikechukwu and A. Ben Abdallah, "Fault-tolerant Spike Routing Algorithm and Architecture for Three Dimensional NoC-based Neuromorphic Systems," IEEE Access, vol. 7, pp. 90436-90452, DOI: 10.1109/ACCESS.2019.2925085, 2019.

[20] Y.-Y. Chen, E.-J. Chang, H.-K. Hsin, K.-C. Chen and A.-Y. Wu, "Path Diversity-aware Fault-tolerant Routing Algorithm for Network-on-Chip Systems," IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 3, pp. 838-849, 2016.

[21] J. Zhou, H. Li, T. Wang and X. Li, "LOFT: A Low-overhead Fault-tolerant Routing Scheme for 3D NoCs," Integration, vol. 52, pp. 41-50, 2016.

[22] Y. Kurokaw and M. Fukushi, "Passage of Faulty Nodes: A Novel Approach for Fault-tolerant Routing on NoCs," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 102, no. 12, pp. 1702-1710, 2019.

[23] R. S. Sutton, "Introduction: The Challenge of Reinforcement Learning," Proc. of Reinforcement Learning, Part of the Springer International Series in Engineering and Computer Science Book Series, vol. 173, pp. 1-3, Springer, Boston, MA, USA, 1992.

[24] S. Mahadevan, "Average Reward Reinforcement Learning: Foundations, Algorithms and Empirical Results," Machine Learning, vol. 22, no. 1-3, pp. 159-195, 1996.

[25] C. J. Watkins and P. Dayan, "Q-learning," Machine learning, vol. 8, no. 3-4, pp. 279– 292, 1992.

[26] F. Farahnakian, M. Ebrahimi, M. Daneshtalab, P. Liljeberg and J. Plosila, "Q-learning Based Congestion-aware Routing Algorithm for on-chip Network," Proc. of the 2nd IEEE International Conference on Networked Embedded Systems for Enterprise Applications, pp. 1-7, Perth, WA, Australia, 2011.

[27] F. Farahnakian, M. Ebrahimi, M. Daneshtalab, J. Plosila and P. Liljeberg, "Adaptive Reinforcement Learning Method for Networks-on-Chip," Proc. of the IEEE International Conference on Embedded Computer Systems (SAMOS), pp. 236-243, Samos, Greece, 2012.

[28] S.-C. Kao, C.-H. Huck Yang, P.-Y. Chen, X. Ma and T. Krishna, "Reinforcement Learning Based Interconnection Routing for Adaptive Traffic Optimization," Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip, pp. 1-2, DOI: 10.1145/3313231.3352369, 2019.

[29] F. Farahnakian, M. Ebrahimi, M. Daneshtalab, J. Plosila and P. Liljeberg, "Optimized Q-learning Model for Distributing Traffic in on-chip Networks," Proc. of the 3rd IEEE International Conference on Networked Embedded Systems for Every Application (NESEA), pp. 1-8, Liverpool, UK, 2012.

[30] F. Farahnakian, M. Ebrahimi, M. Daneshtalab, P. Liljeberg and J. Plosila, "Bi-LCQ: A Low-weight Clustering-based Q-learning Approach for NoCs," Microprocessors and Microsystems, vol. 38, no. 1, pp. 64-75, 2014.

[31] S. Sakamoto, R. Obukata, T. Oda, L. Barolli, M. Ikeda and A. Barolli, "Performance Analysis of Two Wireless Mesh Network Architectures by WMN-SA and WMN-TS Simulation Systems," Journal of High Speed Networks, vol. 23, no. 4, pp. 311-322, 2017.

[32] R. Mohammadi and H. Boroumand Noghabi, "SAT: Simulated Annealing and Tabu Search Based Routing Algorithm for Wireless Sensor Networks," International Journal of Computer Networks and Communications Security, vol. 4, no. 10, Paper ID: 286, 2016.

[33] A. Norollah, D. Derafshi, H. Beitollahi and A. Patooghy, "PAT-Noxim: A Precise Power & Thermal Cycle-accurate NoC Simulator," Proc. of the 31st IEEE International System-on-Chip Conference (SOCC), pp. 163-168, Arlington, VA, USA, 2018.

[34] V. A. M. Catania, S. Monteleone, M. Palesi and D. Patti, "Noxim: An Open, Extensible and Cycle-accurate Network on Chip Simulator," Proc. of the 26th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP), pp. 162-163, Toronto, Canada, 2015.

[35] S. Khan, S. Anjum, U. Ali Gulzari and F. Sill Torres, "Comparative Analysis of Network-on-Chip Simulation Tools," IET Computers & Digital Techniques, vol. 12, no. 1, pp. 30-38, 2017.

[36] I. L. P. Pires, M. A. Z. Alves and L. C. P. Albini, "Trace-driven and Processing Time Extensions for Noxim Simulator," Design Automation for Embedded Systems, vol. 23, no. 1-2, pp. 41-55, 2019.

[37] M.-H. Tang and L. I. N. Jing, "A Quantitative Study on NoC Traffic Scenarios," DEStech Transactions on Computer Science and Engineering, DOI: 10.12783/dtcse/iece2018/26648, 2018.

[38] S. Jog, Z. Liu, A. Franques, V. Fernando, H. Hassanieh, S. Abadal and J. Torrellas, "Millimeter Wave Wireless Network on Chip Using Deep Reinforcement Learning," ACM Conf. on Special Interest Group on Data Communication (SIGCOMM'20 posters/demos), p. 1-3, DOI 10.1145/3405837.3411396, 2020.

[39] Z. Li and Y. Li, "Use Deep Reinforcement Learning for NoC Design," [Online], Available: https://aml-2020.aminer.cn/proposal/ZhiyaoLi_YiweiLi.pdf, 2020.

[40] T.-R. Lin, D. Penney, M. Pedram and L. Chen, "A Deep Reinforcement Learning Framework for Architectural Exploration: A Routerless NoC Case Study," Proc. of the IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 99-110, San Diego, CA, USA, 2020.

[41] S. Deb and H. Kumar Mondal, "Wireless Network-on-Chip: A New ERA in Multi-core Chip Design," Proc. of the 25th IEEE Int. Symposium on Rapid System Prototyping, pp. 59-64, New Delhi, India, 2014.

[42] A. Ganguly, K. Chang, S. Deb, P. P. Pande, B. Belzer and C. Teuscher, "Scalable Hybrid Wireless Network-on-Chip Architectures for Multicore Systems," IEEE Transactions on Computers, vol. 60, no. 10, pp. 1485–1502, 2010.

[43] E. Tahanian, M. Rezvani and M. Fateh, "A Novel Wireless Network-on-Chip Architecture for Multicore Systems," Proc. of the 26th IEEE International Computer Conference, Computer Society of Iran (CSICC), pp. 1-8, Tehran, Iran, 2021.

[44] C. JCH Watkins and P. Dayan, "Q-learning," Machine Learning, vol. 8, no. 3-4, pp. 279-292, 1992.

[45] Md S. Shamim, N. Mansoor, R. Singh Narde, V. Kothandapani, A. Ganguly and J. Venkataraman, "A Wireless Interconnection Framework for Seamless Inter and Intra-chip Communication in Multichip Systems," IEEE Transactions on Computers, vol. 66, no. 3, pp. 389-402, 2016.

[46] K. Gola and B. Gupta "An Energy-efficient Quality of Service (QoS) Parameter-based Void Avoidance Routing Technique for Underwater Sensor Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 05, no. 03, pp 244-261, December 2019.

ملخص البحث:

يمكن استخدام الشّبكات اللّاسلكية على الدارات المتكاملة بديلاً عن تقنية القضبان في الدارات المتكاملة عالية المحاور، وفيها تتمّ الاستعاضة عن المسارات متعددة القفزات بين المحاور البعيدة بعضها عن بعض بروابط لاسلكي ذي قفزة واحدة. إنّ السبب الرئيسي لاستخدام الاتصال اللاسلكي هو التقليل من التأخير واستهلاك الطاقة. ونظراً لتحديد الموارد، فإنّ أداء الشّبكات اللاسلكية على الدارات المتكاملة حسّاس لخوارزمية التّسيير. ففي الوقت الذي تعمل فيه خوارزمية التّسيير الملائمة على خفض التّأخير، فإنّ عليها تجنب الانسداد.

في هذه الورقة، نقترح خوارزمية تسيير مبتكراة باستخدام "تعلُّم كيو"، وهو إحدى طرق تعلم التعزيز لموازنة المرور في الشبكات اللاسلكية على الدارات المتكاملة. فباستخدام هذه الخوارزمية، يمكن للعُقد أن تتخذ القرارات بناءً على ظروف الازدحام في الشبكة عند نقل البيانات من عُقدة المصدر الى عُقدة الهدف. وقد بينت نتائج المحاكاة أن استخدام تعلّم التعزيز بواسطة الخوارزمية المقترحة لتسيير الحُزم يحسّن أداء الشّبكة بشكلٍ ملحوظ؛ إذ تمّ تحسين أداء النّظام بنسبة لا تقل عن 8% مقارنة بأداء الأنظمة المشابهة في أعمال سابقة.

INCREASING SECURITY IN MILITARY SELF-PROTECTED SOFTWARE

Carlos Gonzalez

(Received: 4-May-2021, Revised: 2-Jul.-2021, Accepted: 9-Jul.-2021)

ABSTRACT

The objective of this article is to describe a process methodology to increase security inside secure military self-protected software. Self-protected software is vulnerable to threats, most dependant on the software user. Therefore, detection by self-protected software of the current user is very important. The methodology includes three phases: detection of user, analysis of current state and reaction actions. The detection phase is comprised of assessing geographic location, time at present location and determining user kind (friend or foe). Analysis phase consists of analysing if self-protected software should be at present location, predicting future locations and assessing the location level of threat. Reaction phase includes determining immediate and delay actions if any and perform actions accordingly. Legal concerns are explained, countermeasures and covert actions are proposed and described. An analytical model shows that self-protected software that includes user detection provides more protection than self-protected software without user detection.

KEYWORDS

User detection, Secure military software, Secure self-protected software, Covert actions, International law.

1. INTRODUCTION

One important feature of military secure self-protected software systems is that these systems must be able to detect the current users of the system. This article describes a methodology and provides a list of useful guidelines to obtain user detection for military secure self-protected software systems, the analysis of the software current state and possible reactions to threats.

It is important to point out that this article does not include or discuss the use of covert electronic surveillance of any kind inside the self-protected software described in this paper [1].

The types of military secure software self-protected systems addressed in this paper range from the military software application running in a user laptop or cell phone, to the ones that generally are not connected to a network of any kind and/or are embedded software running on a specific device. Following is description previous research in areas related to this work.

1.1 Intrusion Detection Systems

Much work has been done on network and host intrusion detection systems to identify user-intruders in networks.

Amer et al. [2] provided researchers with a taxonomy and survey of current dataset composition and current Intrusion Detection System (IDS) capabilities and assets. These taxonomies and surveys aim to improve both the efficiency of IDSs and the creation of datasets to build the next-generation IDSs as well as to reflect networks threats more accurately in future datasets. To this end, this manuscript also provides a taxonomy and survey of network threats and associated tools.

Lunt [3] described the use of such tools for detecting computer system intrusion and describes further technologies that may be of use for intrusion detection in the future.

Zhang et al. [4] examined the vulnerabilities of wireless networks and argued for the inclusion of intrusion detection in the security architecture for mobile computing environments. They developed an architecture and evaluated a key mechanism in this architecture by anomaly detection for mobile *ad-hoc* network through simulation experiments.

Koch et al. [5] presented a model-driven approach to engineer self-protection for autonomous systems. The approach is integrated into model-driven security SecureUML for modeling access control and

C. Gonzalez is with the Facultad de Sistemas, Universidad Autonoma de Coahuila, Mexico. E-mail: gonzalezc757@gmail.com

supports the system designer in engineering self-protection rules to react to unexpected security vulnerabilities. Self-protection is specified by a set of transformation rules which restrict the security model. A graph-based semantics for the transformation rules allows to verify that security requirements are satisfied by the specified self-protection rules.

As the deployment of network-centric systems increases, network attacks are proportionally increasing in intensity as well as in complexity. Attack detection techniques can be broadly classified as being signature-based, classification-based or anomaly-based. Nashif et al. [6] presented a multi-level intrusion detection system (ML-IDS) that uses autonomic computing to automate the control and management of ML-IDS. This automation allows ML-IDS to detect network attacks and proactively protect against them. ML-IDS inspects and analyzes network traffic using three levels of granularities (traffic flow, packet header and payload) and employs an efficient fusion decision algorithm to improve the overall detection rate and minimize the occurrence of false alarms.

Elkhodary et al. [7] developed a methodology for designing security and designing security-aware adaptive systems that adapt in response to an attack. Adaptive application security requires all reconfiguration scales, automated detection and resolution of conflicts and the consideration of security features collectively.

Ankit Thakkar et al. [8] in their study reviewed the datasets developed in the field of Intrusion Detection Systems (IDSs). These datasets have been used for performance evaluation of ML- and DM-based IDSs. The study revealed that there is a need to update the underlying dataset to identify the recent attacks in the field of IDSs with improved performance.

Kyle Guercio [9] guide covered the industry’s leading Intrusion Detection and Prevention Systems (IDPS), along with a summary of key features to look for as you evaluate solutions.

1.2 Software and Hardware Specifically for Military

Much software and hardware work has been done specifically for military. Here are some examples of unclassified work.

Mahmoud et al. [10] reported information on network security in the aeronautical communication domain. Three fundamental research axes are explored. First, a quantitative network security risk assessment methodology is proposed. Their approach is based on the risk propagation within the network nodes. As study cases, the algorithm has been validated in the scope of the European industrial project entitled SESAR (Single European Sky ATM Research) and the Aerospace Valley FAST (Fibrelike Aircraft Satellite Communications).

Keller [11] reported that BAE Systems received a $4 million contract from the Navy for a quick-turnaround demonstration of a new radio frequency countermeasure system for the P-8A Poseidon aircraft. The electronic warfare (EW) system will be a lightweight pod mounted to the aircraft that will add self-protection capability to the Poseidon. It consists of a small jammer, a high-powered amplifier and the AN/ALE-55 Fiber-Optic Towed Decoy and lures enemy missiles away from the aircraft to the towed decoy.

Will developing intrusion detection capabilities meet the operational, performance and implementation goals of the US Air Force? To help ensure that they will, the MITRE C2 Protect Mission-Oriented Investigation and Experimentation (MOIE) project has been focusing on Air Force needs with a view to articulating them to commercial interests that may develop capabilities. This may help shape future funding decisions and may also provide a common framework for discussing issues.

LaPadula [12] explained a first cut at defining goals, capitalizing on customer and corporate experience with intrusion detection tools as well as knowledge of the problem domain. They created an information base about intrusion detection, providing a framework for discussing, refining and enhancing intrusion detection goals. One technological countermeasure is intrusion detection capability. Once detected, a variety of actions can be taken to thwart an attacker’s intentions. In the recent past, intrusion detection capabilities have been developed by both governmental and commercial interests.

Passive IDS includes such components as security cameras, motion detectors, thermal imaging systems, radar, card readers and keypad systems that are designed to monitor and limit entry to facilities. These systems are thought of as a kind of firewall used to keep out unwanted guests. Just like systems designed for online protection, IDSs have grown more automated over time with the addition of such things as voice recognition, facial recognition, license plate scanners and more.

Andone [13] explained reactive systems, also known as Intrusion Prevention Systems (IPSs), where the ability to respond in kind to suspicious activity is hardwired into the matrix. Like watch dogs, these systems not only observe, but also are tasked with providing feedback to system operators and security personnel.

1.3 Emerging Technologies

The methodology described below can be used for the creation of Runtime Application Self-Protection (RASP) applications, which represent an emerging security technology [14]. As described in Veracode [15] RASP resides on the server and embeds security into the running application. It then intercepts all calls to the system and ensures that it is secure by implanting validation of data requests directly into the application. RASP can be applied to Web and non-Web applications and does not affect the application design. Rather, the detection and protection features are added to the server on which an application runs. With RASP, enterprises are not building a secure application, but rather adding a "shield" to the code. If an application is defective, it will remain so even when protected by RASP. In other words, RASP is not going to make an app as secure as it would be if all the security requirements for the self-protected software were built into the app from start to finish. The methodology described below carries the advantage of the security built into the code from the start.

2. ASSUMPTIONS

For security reasons in this paper when we talk about an algorithm, procedure or methodology, we describe only the function's general purpose and how it is expected to work. We do not give details of the algorithms, procedures or methodology, nor how these should be implemented.

We also assume that for the military secure self-protected software system, including the algorithms and procedures described here, the only code available to the user will be the executable code.

We also assume that the design and implementation of the self-protected software include from inception in its specifications and requirements all the security needed for the self-protected software.

3. METHODOLOGY

It is important to clarify that the user detection function addressed in this methodology is not the detection of users coming through a network, but the actual user using the self-protected software.

The proposed methodology approach describes a process to increase security inside secure military self-protected software. Self-protected software is vulnerable to threats, most of which depend on the software user. Therefore, detection by self-protected software of the current user is very important. The methodology approach includes three phases: detection phase of user, where a process methodology is used to best determine physical location and the user kind (friend or foe) executing the self-protected software; the analysis phase, in which a rigorous analysis is done of the current user and state of the self-protected software; and a reaction phase where, given the current kind of threat taking place, a set of actions are performed. The complete architecture of the methodology process is presented in Figure 1.

3.1 Detection Phase

The detection phase is comprised of the following processes; assessing geographic location, time at present location and determining user kind. Figure 2 shows the flow diagram of the software processes comprising this phase.

3.1.1 Assessing Software Geographic Location

A key issue that self-protected software should be aware of is the current physical location of the device. To be a truly self-protected software, this software should be aware of its immediate surroundings/ environment. The environment will be different in friendly *vs.* enemy territory.

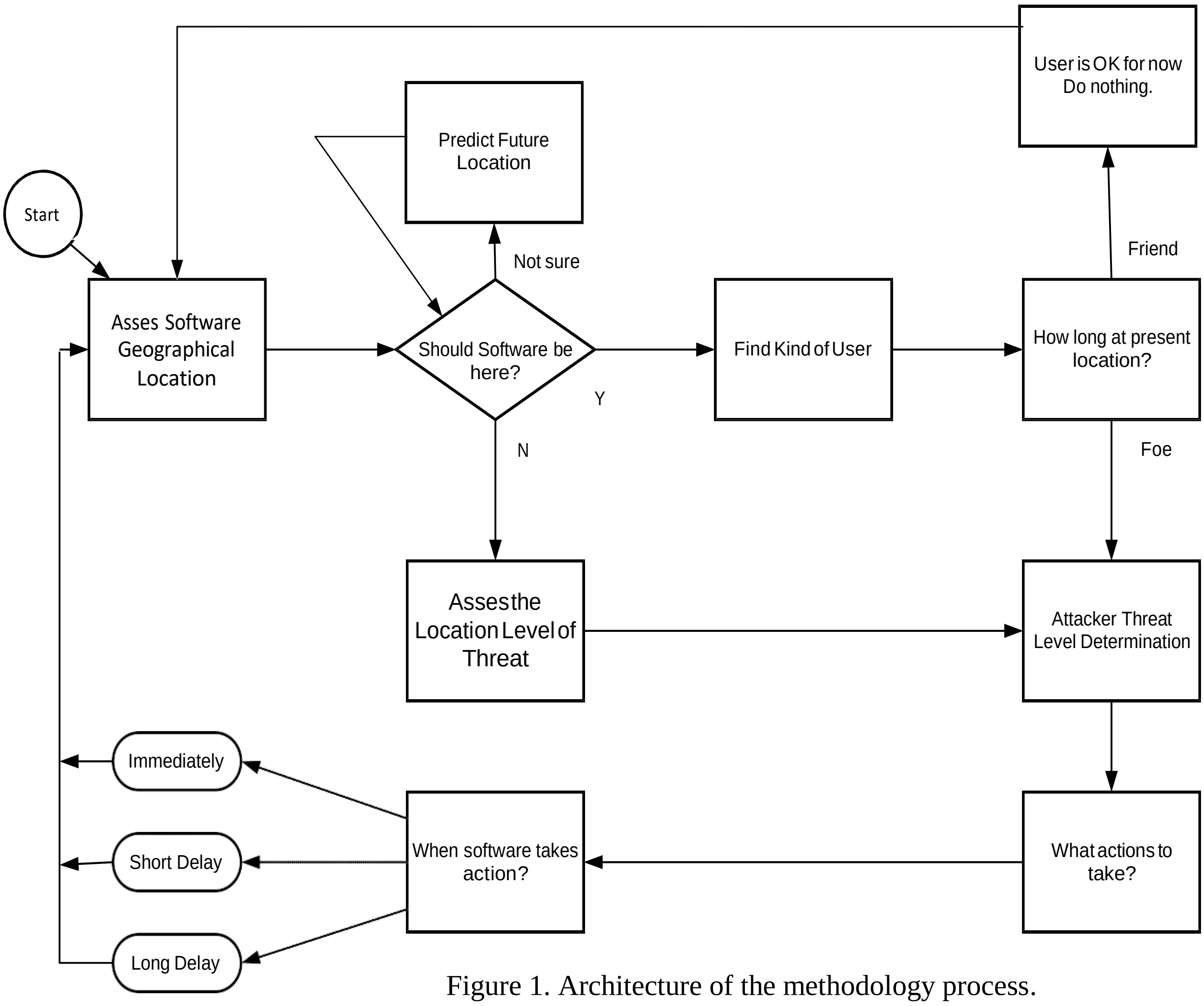

Start
Asses Software Geographical Location
Predict Future Location
Not sure
Should Software be here?
Y
N
Find Kind of User
How long at present location?
Friend
User is OK for now Do nothing.
Foe
Asses the Location Level of Threat
Attacker Threat Level Determination
What actions to take?
When software takes action?
Immediately
Short Delay
Long Delay

Figure 1. Architecture of the methodology process.

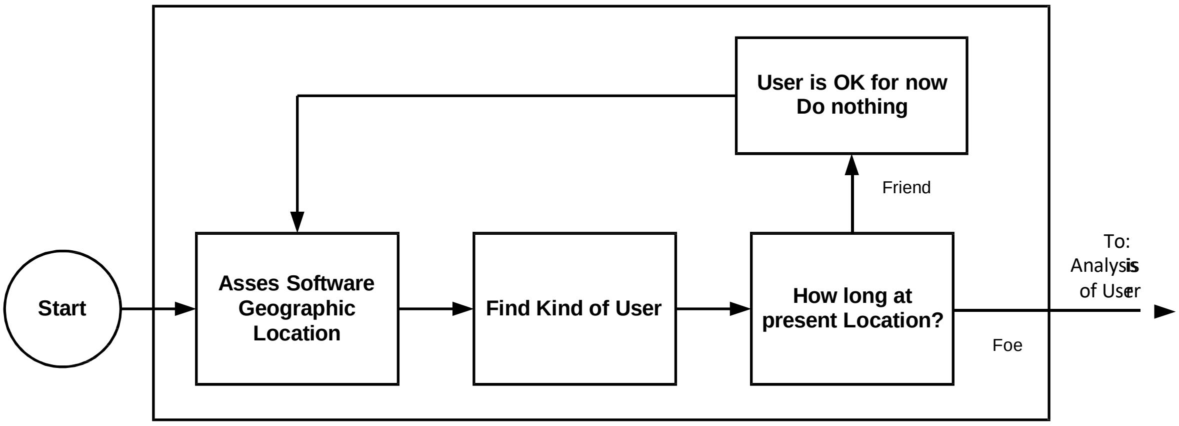

Start
Asses Software Geographic Location
Find Kind of User
How long at present Location?
Friend
User is OK for now Do nothing
Foe
To: Analysis of User

Figure 2. Detection phase.

The software self-protected actions and responses will vary according to the software current location.

Since the software's current location may change over time, one of the most important questions that one should ask is: Where is the software now? This basic question would not be easy to answer if the software does not have some sort of GPS.

If the original manufacturer is providing the hardware and the software, then the author of this article suggests that the following devices be included inside the hardware of a secure self-protected software system:

(1) At least two sets of GPS devices available to the software.

(2) At least one Wi-Fi mechanism to connect to the internet.

(3) At least one Bluetooth mechanism to receive and send signals.

(4) At least one mechanism to destroy our local hardware (See note below).

(5) An explosive to destroy an area in close proximity to our hardware (See note below).

(6) At least one heartbeat mechanism to monitor the devices installed in the hardware in order to ensure that all are working.

It is important to point out that item number 4: the mechanism to destroy the local hardware should exist only when the hardware is considered a specialized hardware (e.g. a sonar system, a tactical radio or a fire control system). It does not apply to cell phones or regular laptops. In such instances, when the hardware is not considered specialized, it would be better to have implemented a "disable" mechanism instead of a "destroy" one. Therefore, the software will have the "destroy" mechanism when running on an embedded system part of any military hardware. In order to avoid false-positives, it is recommended to have fail-safe mechanisms to ensure that when such drastic action is required; for example, communicate to the user when that what he/she is doing is illegal with consequences. If the warning (may be more than one) is not taken, then the destroy action will commence.

Also, on item number 5, an explosive to destroy an area in close proximity to our hardware having this option should be exercised only when dealing with highly sensitive software/hardware that is important to national security. The author firmly believes that in some cases, we have to react as if we were at war. In other words, we do not ever want to be in the hands of a foreign country that is our known foe. Again, as stated in item number 4, in order to avoid false-positives, it is recommended to have fail-safe mechanisms to make sure that such drastic actions are required. One such action could be a warning to the user, but if our intent is to continue in stealth mode, another action could be degrading the performance of the system and seeing how the user reacts to this action.

If the software has access to a GPS, the software should know in which country and city or region of the world the software is (longitude and latitude). The software should take a couple of GPS time measurements to determine whether it is moving and the speed of its movement.

If the software is able to determine all the information described above, then it can continue with assessing the kind of user (see step 3.1.2).

If one of the GPS devices is not working, it is a matter of concern depending on where the software is, but if all the GPS devices are unavailable, then this is a sign of a possible threat. It is possible that the software is in friendly territory, but in a location without GPS signal. The software should then analyze the previous GPS locations and their time of occurrence. If the previous analysis of the user was determined to be a friendly user or at a friendly location, then the software does not have to take any immediate action, but should stay alert. If on the other hand the previous detected user was non-friendly or at a non-friendly location, this should be considered as a possible threat and it should be acted accordingly.

If GPS is initially not available with the secure self-protected software, then the software must use other means to determine the location and type of user.

(1) The first option is for the self-protected software to go and look into the user file system (if such file system is available to our software).

    a. The names of the files could be a good lead to the user's nationality (i.e., files with mostly Chinese names indicate with high probability a Chinese user).
    b. The contents of the files may also lead to the user's nationality.

(2) If the software is running on a phone or laptop, ask the user to identify himself/herself.

    a. Ask for the user password for the use of the device.
    b. If the device has any biometric capture means, ask to provide such biometric data.

(3) Try the user's internet connection if available, to gain access to user emails.

    a. The contents of the e-mails may help identify the user's nationality.
    b. The headers of the e-mail may give the location of the user.

(4) If a direct internet connection is available, a message should be sent to a specific recipient to include all the captured information about the current location and the type of user threat. Software will delete all traces of the sent e-mail. Even when an answer is not expected back, the software has made the home base of the software aware of the situation and has given as much information as possible to be used for possible external actions or to improve the next generation of self-protected software.

(5) If a direct internet connection is not available, software should try to see whether it could establish communications through Wi-Fi.

(6) Software should try to communicate with any devices in close proximity *via* Bluetooth. If software can get into user's cell phone, software can obtain much information from this source.

(7) Software should analyze all the devices connected to software's computer [16].

a. The type of computer used may give software information about the user.
b. The type of keyboard used may lead to the user's nationality.
c. The type of printer. For example, a printer printing in Arabic is a good indication of the user's nationality or origin.

Note: All communications, searches and testing should be done in stealth mode; that is, no lights, noises or movements should be produced by these actions or at least they should be minimized.

Another form of attack is a GPS spoofing attack. These attacks attempt to deceive a GPS receiver by broadcasting incorrect GPS signals. The signals are structured to resemble a set of normal GPS signals or rebroadcast genuine signals captured elsewhere or at a different time. Even though the military GPS P-code is heavily encrypted and the Y-code encryption algorithm is not available to civilian users and would be difficult to spoof, the software should always monitor the GPS signals to detect and prevent spoofing. Wen et al. [17] described 10 different procedures to prevent GPS spoofing.

3.1.2 Attacker Threat Level

We use the four types of intruders defined in Gonzalez [18] and listed here:

"Casual-attacker: The attacker is not technically knowledgeable enough to do any reverse engineering and retrieve data or algorithms from the machine code software.

Hacker-attacker: This attacker is enough knowledge to access and retrieve some data and/or algorithms from the machine code sources of the software. Code development of the self-protected software should as a minimum include security procedures for attacks of this kind.

Institution-attacks: These are attacks sponsored by industry. They either hire somebody or use own resources to do the attack. This kind of attack is called industrial espionage.

Government-attack: The government of a nation through one or more of its agencies generates this attack and will use all the technical and legal resources available to such agencies."

It is important to notice that the main difference between the Gonzalez [18] previous work and this work is that in the previous work, there is Internet access by the software and a secure server exists to communicate with the standalone software, which is not the case with this current work, in which an Internet access is not needed and there is no secure server communication with the standalone software.

To determine the kind of threat the current user represents, the software has to evaluate all the information available about the user and the current location. Following is a list of situations that help the software determine the level of user threat:

(1) When the software detects that it has been modified (i.e., GPS routine was bypassed), then the threat is at least a hacker-attack. It is highly recommended to have at least two different locations inside the software where the software can check this detection of software modification.

(2) The software’s current location indicates that the level is an institution attack or government attack. It is sometimes very difficult to differentiate between institution-attack level and government-attack level. Therefore, if this software is expected to survive a level of government attack, then even if the software has a level of institution attack threat, it should treat it as a level government-attack. If on the other hand the maximum level the software is trying to protect against is the level of institution-attack (not a national security threat), then the self-protected software reaction to the threat could be economically based (see step 3.3).

(3) When the computer running the self-protected software is connected to any foreign device (see step 3.1.1), then the threat should be an institution attack or a government attack.

(4) When the system has a heartbeat device and this device reports an anomaly, the threat should be an institution attack or a government attack.

(5) When the self-protected software is at a non-friendly location, the minimum assumed threat level is a hacker attack. The self-protected software may be designed to differentiate between non-friendly, bad-non-friendly and very-bad-non-friendly attacks and set the threat accordingly.

(6) When the self-protected software is at a friendly location, with no signs of tampering, then the user can be considered friendly and defined as a no-threat at this time.

3.1.3 How Long at Present Location?

The amount of time in the current location is very important.

(1) When the software has been at the current location for some time (e.g. a month) and it is a friendly territory and the user is a non-threatening user, then the software can be assumed to be currently safe.

(2) If the location is new or is a recent location, the software has to continue to analyze the user.

(3) If it is an old location, but the location and/or user were determined to be a threat or a possible threat, the software should then decide whether the time has come to take extra measures with this user (see step 3.2).

3.2 Analysis Phase

Analysis phase consists of analyzing whether the software should be at the present location, predicting future locations or assessing the location level of threat. Figure 3 shows the flow diagram for this phase.

3.2.1 Should the Software Be Here?

If the software has the option of handling missions, then the question to ask is whether the software is inside the mission parameters (i.e., the geographical area designated for the mission) or not. If the software does not have the mission option, then once the software has determined its location (as accurately as possible), the question is: Should the software be here?

(1) If the answer is no, then this is a bad location and the software must immediately continue to determine the attacker threat level (see step 3.2.2). The software may be at the right location (example, city and country), but the user may still be a foe.

(2) If the answer is yes, the software may be in the right location (example, city and country), but the user may still be a foe. With a foe user, the software must immediately asses the level of threat and act accordingly (see step 3.2.2). A friendly user in a good location takes the software to the beginning of the self-protected loop.

(3) There may be another answer for the posted question: the software is not sure. This may be possible for several reasons. The software may be on a ship and now it is on international waters. The software does not know if the ship is a friend or a foe. The software must investigate more before taking any action.

If the software is moving, it should obtain the rate of change (e.g., walking, car, ship, airplane, …etc.) and depending on the type of rate, it should determine the direction the software is heading. With the heading and the rate of change, the software will generate a set of predicted future locations. Then, the software should analyze some of the predicted future locations and take action depending on where these locations are and how soon the software is going to get there.

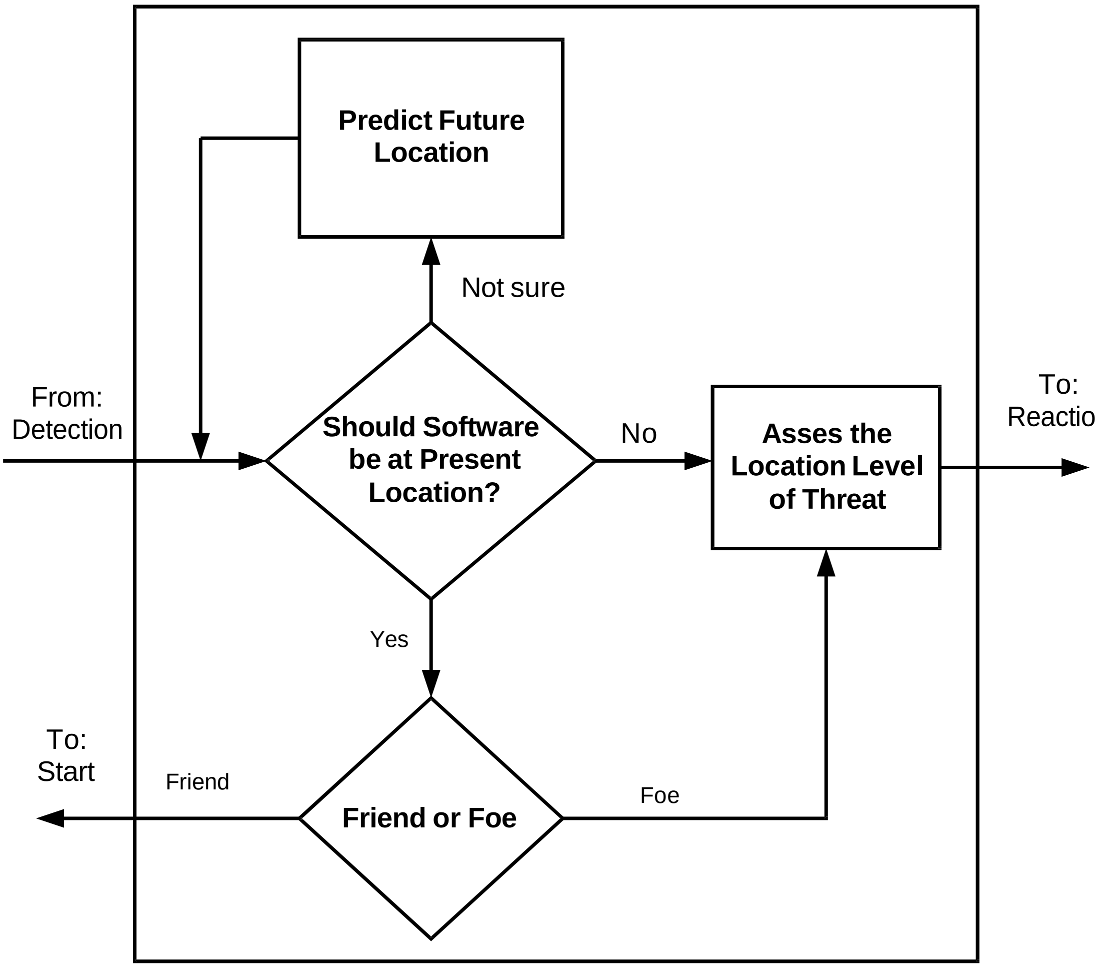


Figure 3. Analysis phase.

3.2.2 Assessing the Location Level of Threat

The question now is: How big of a threat is this current location or expected location in the near future?

At this point, the software may receive the following three kinds of scenario:

A friendly user in an unfriendly location. For this case, the self-protected software must stay alert, do nothing, but keep vigilant. An example of this kind of scenario is if the software is made in the U.S., then a friendly user may be an Australian user in a Venezuelan location.

A foe in a friendly location and a foe in an unfriendly location. For both cases of foe users, the software must go to the next phase and decide what to do next (step 3.3).

3.3 Reaction Phase

Reaction phase includes determining what actions to take and when to take such actions if any and finally perform actions. Figure 4 shows the flowchart of the process in this phase.

3.3.1 What Actions Should the Software Take?

The actions to take are going to change depending on the location, friend or foe and level of threat.

For a friend on a friendly location, only a signal is sent reporting any threat detected. History of threat and actions taken are saved.

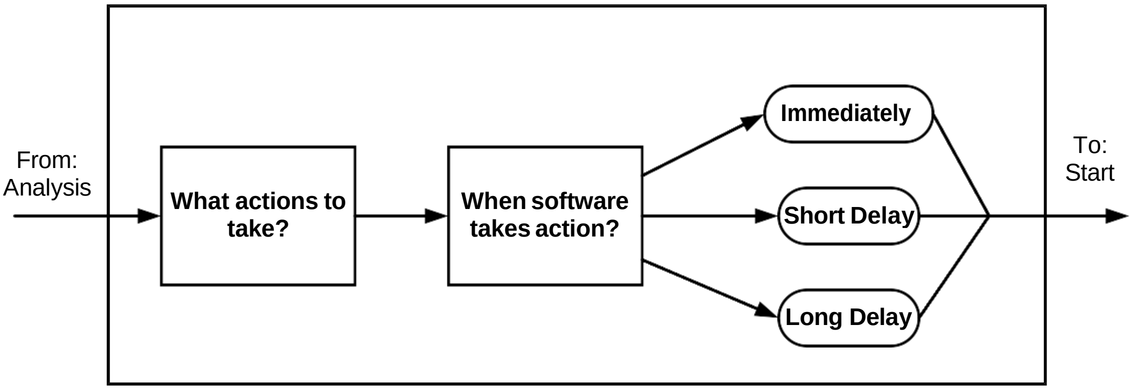


Figure 4. Reaction phase.

A friend on an un-friendly location should make the software heighten its awareness and make the software take more measurements (could be to add different ones or just increase the rate of taking measurements). This situation is not stable and can change very fast.

With any foe, the software should assess the level of threat and act accordingly.

For a casual attacker at any location, the author recommends taking all or some of the following actions:

(1) Delete and erase all files related to the self-protected software. If this is not possible, the system should try to re-encrypt all the self-protected software.

(2) If Internet is available, send a signal home (home, being a place in the Internet designed and operated by the government agency owner of the self-protected system) reporting the findings and actions taken.

(3) Erase or corrupt most of the user files.

(4) Display a message to the user stating that a malicious virus has taken control (the idea is to mislead him/her on the source of the problem and actions taken).

For a hacker attack when the software is at a friendly location and is guarding a maximum hacker attack, the author recommends taking all or some of the following actions:

(1) Perform same actions 1-3 of the casual attacker.

(2) Display a strong message to the user saying that his/her actions have been reported to the FBI, CIA, Interpol, …etc.

If we have the case that the software has a hacker attack and it is guarding against an institution attack or a government attack or the software is at an un-friendly location, the author proposes that the system performs the following actions and such actions be executed covertly.

(1) Modify and change the self-protected software (or its data) slightly (these actions should be decided and determined at the design time of the secure self-protected software); so it produces results, but the wrong results.

(2) Insert a malicious virus that spreads to all the contacts of this user and all nodes of the network used by this computer.

(3) Start the processes to destroy the local equipment. The due time could be up to a couple of days in order to give time in case the status of the threat changes, like moving to a friendly location. If a physical bomb is not possible (because of physical limitations or the law does not permit such action), then at the due time destroy as much software and information as possible. The destruction of the local equipment should never include human lives. Before any destruction is done to software, equipment or general area, it is necessary to be sure that this is not a false-positive. It is recommended to have fail-safe mechanisms to make sure that when such drastic action is required, we have a high probability of certainty.

For an institution attack (all covert actions):

(1) At friendly locations: Same actions 1-3 of casual attacker.

(2) For un-friendly locations: Same as hacker attack at an un-friendly location, except that the physical bomb due time is extended for a longer period (six months to a year). The idea is to do as much damage as possible.

For a government attack (all covert actions):

(1) Same actions as 1-3 of casual attacker.

(2) Insert a malicious virus that spreads to all the contacts of this user.

(3) Detonate a software bomb to destroy as much software and information as possible.

(4) Detonate a physical bomb for the destruction of the local equipment only. If this is a guard for government attack, this option should always exist when the software is delivered with the hardware included.

(5) Detonate a physical bomb for a larger area at the current location. This action may end up costing human lives. There is always the option of giving a warning to the current user before the action is taken. The decision to have this option installed in the system will be up to de owner of the software. This may not always be possible if no hardware is delivered with the software.

In all cases, the system should be aware of the legal principle of ‘fair labelling’ [19], which requires that the label of the offence (e.g. braking and modifying the self-protected software) should fairly express and signal the wrongdoing of the accused, so that the stigma (and receiving actions) of conviction corresponds to the wrongfulness of the act.

3.3.2 When Should the Software Take Action?

If the level of threat is determined to be a government attack, then, independently of the location, the actions to take must be immediate.

If the level of threat is an institution attack and the software is at a non-friendly location, then the actions will be taken immediately. On the other hand, if the software is at a friendly location, the author suggests a delay of the software bomb deployment.

For casual attacker and hacker-attack threat levels at friendly locations, the actions are immediate. For hacker-attack threats in non-friendly locations, the author suggests the delay of the software attack to the local computer for a couple of days, in order to make sure that the location is correct or to make sure that the threat level doesn’t increase.

If the level of threat is an institution attack and the software is at a friendly location, the actions are taken immediately. For non-friendly locations, it is suggested for example to delay in the software bomb deployment.

4. LEGAL ASPECTS

Over the last twenty years, many international lawyers have successfully crafted an elaborate and operational system of international criminal law [19]. The project used precepts of criminal law, international human rights and humanitarian law. The last two areas provided essential normative content and a familiar framework for international oversight and intervention.

Like most international issues, each country has its own way of interpreting the international law, especially when dealing with cyber war, covert actions or surveillance. In a major story [20], Yahoo News disclosed the existence of a 2018 presidential covert action finding altering the terms on which the CIA can (and should) engage adversaries *via* cyber means. The story is an important reminder that the CIA continues to play a critical role in the increasingly fierce gray zone competition that characterizes statecraft in cyberspace these days.

One of the most interesting points in the article involves a rule change that apparently removed or weakened a prior prohibition on operations that might “damage critical infrastructure.” Thus, a blank check to blow up critical infrastructure would indeed be deeply concerning. But it is possible—indeed, probable—that the finding does not give such a blank check. Possibly it just authorizes prepositioning of capabilities, in the event an adversary takes such an action against U.S. critical infrastructure (in our case it will be against the self-protected software) and thus opens the door to such a countermeasure.

At the current time [21], the development of information warfare presents international legal issues that will complicate nation's efforts both to execute and to respond to certain information warfare attacks, specifically those using computers, telecommunications or networks to attack. We have described actions to take when the self-protected software is under attack, but the legality of such actions is left to the owner and designer of the self-protected software.

Information warfare weapons must meet the same tests for necessity and proportionality as other weapons under the laws of armed conflict [22]. In addition, commanders must recognize and weigh the possible consequences of weapons that can devastate the information systems of an adversary. Problems such as lack of enemy command and control, post-hostility reconstruction and retaliation, among others, must be considered by the commander contemplating the use of information weapons. Because of the extraordinary consequences of these weapons, developers must provide guidance for their employment and commanders must carefully consider adverse effects from their use.

In the field of cyber security, ill-defined concepts and inconsistently applied terminology are further complicating an already complex issue [23]. This causes difficulties for policy-makers, strategists and academics. Using national cyber security strategies to support current literature, this paper undertakes three tasks with the goal of classifying and defining terms to begin the development of a lexicon of cyber security terminology. The term “active defence” is common in the military as the idea of offensive action and counterattacks to deny advantage or position to the enemy. The concept remains elusive when applied to the cyber domain and suffers a lack of clarity in related law and national policy.

We want to include in this section issues related to the international law interpreted by Russia and China.

To understand how the Russians interpret international law, we recommend the book by Mälksoo [24]. This book addresses a simple question: how do Russians understand international law? Is it the same understanding as in the West or is it in some ways different and if so, why? It answers these questions by drawing on from three different yet closely interconnected perspectives: history, theory and recent state practice.

For the Chinese interpretation of international law [25], this article highlights a different set of elements that become manifest in assessing the rapid overall rise in references to and application of international law by courts in China in recent years.

5. Analytical Model

Let S be the software that is protected without user detection

Let SD be the software that is protected including user detection

Thus SD=S + {user detection}

Let L be an intruder type with possible values of L1 for Casual-attacker or L2 for Hacker-attack or L3 for Institution-attack or L4 for Government-attack intrusion types

Then L=L1 ∧ L2 ∧ L3 ∧ L4

EF(S,I) the effort (in man-hours) for intruder I to break the software protection and reverse engineer the software S where I \in L

In general, we define EF(S,L1) > EF(S,L2) > EF(S,L3) > EF(S,L4)

Theorem: $\boldsymbol{EF(S,I)\ \leq\ EF(SD,I)\ \ \forall I\ \in L}$

Proof:

Let us assume that a user (intruder) is trying to reverse engineer the software

The effort (in man-hours) used trying to break software S by intruder I at time t is EB(S,I,t)

$$\text{BREAK(S,I,t)}=\begin{cases} true & \text{if software S was broken by intruder I at time t} \\ false & \text{otherwise} \end{cases}$$

Let's call t_b the time at which BREAK(S,I,t)= true

and t=0 the time at which the intruder started trying to break the software.

Thus **EF(S,I)=** $\sum_{t=0}^{t=t_b} \mathbf{EB(S,I,t)}$

If we have user detection, then we can assume then at some time t, t>0 our user detection algorithm detects the type of user and determines the intrusion type

Let us call this time of user detection t_d

We have three options:

$t_b > t_d$,

$t_b = t_d$

$t_b < t_d$,

For $t_b > t_d$, at time t_d BREAK(SD,I, t_d)= false, then at this time our user detection algorithm will take an action accordingly to the type of intrusion threat. This action will by definition make the effort of the intruder to obtain a software break much greater.

Thus EF(S,I) < EF(SD,I)

For tb = td, the software is broken at the same time of the user detection. This will be a low probability type of occurrence because of the exactness of the equality.

Thus EF(S,I) = EF(SD,I).

For $t_b < t_d$, the software was already broken by the time of the user detection. This is the worst case, since the intruder was able to reverse engineer our software. With user detection, we will be able to retaliate for this break and cause some damage to intruder I. We expect that any repairs to the damage caused by our user detection algorithm will generate an effort for the intruder.

Thus EF(S,I) = EF(SD,I)

If we add the effort to repair the damage, then we will add a positive number to our EF(SD,I); that is:

The real EF(SD,I) = old EF(SD,I) + {effort to repair retaliation}

And since the {effort to repair retaliation} is always a positive number, then

EF(S,I) < EF(SD,I).

We have shown that under all circumstances, the final effort to break the software is equal or greater if we use user detection inside our software.

This proves our theorem Q.E.D.

Thus, having user detection inside our software will always be better than not having it.

6. Recommendations and Conclusions

6.1 Main Conclusion

We have proved and concluded that having and using user detection in self-protected software is a means to increase the security of such software. Thus, the main contribution of this article is the description of a process methodology to follow when designing military secure self-protected software. The set of guidelines described in this paper to detect location, size of threat and user type are not meant to be exhaustive.

6.2 Recommendations and Discussion

The topic of this paper is highly sensitive for military; therefore, almost all research papers on this topic are classified as secret or top secret. Currently, there is nothing in the un-classified literature about this topic.

The set of actions to take in response to the different threats to some people may seem to be over the top, but the author firmly believes that in some cases, we have to react as if we were at war. If what we are protecting is of high security value and it will be very negative for our country to have our enemies having or using this Self-Protected Software System, then our actions reflect this belief. Our enemies will not hesitate to illegally use our resources even if this causes economic loss or loss of human lives. This set of actions is defined at the start of our software development life cycle by the owners of the software, which at the end are responsible (morally, economically and legally) for the actions taken by the secure self-protected software. On the other hand, we do not want our software to make mistakes, giving us false-positives and doing unnecessary harm.

We recommend the use of safety standards and methodologies for the development of this self-protected software [26]-[29].

It is important to point out that because of the inherent mobility of cell phones and laptops, the handling of self-protected software security should involve continuous authentication of the user's mobile device [30] along with user detection. It is more probable that a change of user occurs on a mobile device than any other device, because mobile devices are easy to steal, borrow, forget, misplace, …etc.).

When analyzing the user's threat level which is related to the software risk-level:

Casual attacker: The software is only protected for casual attackers. A minimum of security is needed.

Hacker attack: The software is protected against hacker attacks. This intruder may or may not have initial economic gain plans for the intrusion. In most cases, it is the intellectual challenge that motivates this intruder (i.e., hacker), but economic gains may not be very far behind.

Institution attack: The software is protected against institutional attacks. Economic gains are the main reason for the intrusion. In most cases, with enough time and money, any secure self-protected software may be cracked. Therefore, the developing team should always work with the goal of making the intruder's effort needed to break the secure code high enough and not cost-effective.

Government attack: The software is protected against government attacks. In most cases, with enough time and money, any secure self-protected software may be cracked. It is recommended that techniques for intruder detection [31], [3], [6], [32]-[34] (if an Internet connection is available) and the user detection described in this paper with the respective actions to take (covert and not-covert) be included in the secure code. This level of protection requires the use of the most sophisticated security algorithms.

Our software has to be aware of the level of threat it is supposedly guarding against. For hacker attack and institution attack, the main objective of the threat is economic. Therefore, our reaction to a threat of this kind could end up causing the user economic loss. These actions could vary from erasing all software from the machine to releasing a malicious virus to providing wrong output in the software output.

In general, to combat secure self-protected software threats, organizations must adopt pre-emptive and reactive measures by building self-defending mechanisms into their software, so that their application is self-protected on the device. Finally, to make the security of the software complete and more robust, we should obfuscate [35] our secure self-protected software.

The protection and security of self-protected software constitute an ever-present concern for the military agencies of any government producing their own hardware-software military equipment that is mobile. Therefore, the techniques described in this paper should be required reading for industries producing for the military any self-protected software.

For our future work, we are planning to do a simulation of the suggested methodology and we will use AlSobeh [36] framework to do the simulation.

REFERENCES

[1] A. Deeks, "An International Legal Framework for Surveillance," Virginia Journal of International Law, HeinOnline, 2014.

[2] S. H. Amer and J. A. Hamilton, "Intrusion Detection Systems (IDS) Taxonomy: A Short Review," Journal of Software Technology, vol. 13, 2010.

[3] T. F. Lunt, "A Survey of Intrusion Detection Techniques," Computers & Security, vol. 12, no. 4, pp. 405–418, 1993.

[4] Y. Zhang, W. Lee and Y. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks," Mobile Networks and Applications (Georgia Institute of Technology), pp. 1–16, [Online], Available: http://wenke.gtisc.gatech.edu/papers/winet03.pdf, 2003.

[5] M. Koch and K. Pauls, "Engineering Self-protection for Autonomous Systems," Proc. of the International Conference on Fundamental Approaches to Software Engineering (FASE 2006), Part of the Lecture Notes in Comp. Sci. Book Series, vol. 3922, pp. 33-47, DOI: 10.1007/11693017_5, 2006.

[6] Y. Al-Nashif, A. A. Kumar, S. Hariri, G. Qu, Y. Luo and F. Szidarovsky, "Multi-level Intrusion Detection System (ML-IDS)," Proceedings of the IEEE International Conference on Autonomic Computing (ICAC'08), pp. 131-140, Chicago, USA, 2008.

[7] A. Elkhodary and J. Whittle, "A Survey of Approaches to Adaptive Application Security," Proceedings of the Workshop on Software Engineering for Adaptive and Self-managing Systems (SEAMS'07), DOI: 10.1109/SEAMS.2007.2, Minneapolis, USA, 2007.

[8] A. Thakkar and R. Lohiya, "A Review of the Advancement in Intrusion Detection Datasets," Procedia Computer Science, vol. 167, pp. 636–645, 2020.

[9] K. Guercio, "Best Intrusion Detection and Prevention Systems for 2021: Guide to IDPS," eSecurityPlanet, [Online], Available: https://www.esecurityplanet.com/products/intrusion-detection-and-prevention-systems/, 2021.

[10] M. S. Ben Mahmoud, N. Larrieu, A. Pirovano and A. Varet, "An Adaptive Security Architecture for Future Aircraft Communications," Proceedings of the 29th Digital Avionics Systems IEEE Conference (DASC), DOI: 10.1109/DASC.2010.5655363, Salt Lake City, USA, 2010.

[11] K. John, "BAE Systems to Install Electronic Warfare (EW) Self-protection Pod to Help Defend P-8A Poseidon Aircraft," Military & Aerospace Electronics, [Online], Available: https://www.militaryaerospace.com/communications/article/14195763/electronic-warfare-ew-aircraft-selfprotection, 2021.

[12] J. LaPadula Leonard, "Intrusion Detection for Air Force Networks," Mitre Technical Report, MTR 97B0000035, October 1997.

[13] A. Jay, "Intrusion Detection Systems: The First Line of Defense," SCIF Global Technologies, [Online], Available: https://scifglobal.com/intrusion-detection-systems-the-first-line-of-defense/, 2015.

[14] J. P. Mello Jr., "What is Runtime Application Self-protection (RASP)?," TechBeacon, [Online], Available: https://techbeacon.com/security/what-runtime-application-self-protection-rasp application-self-protection-a-must-have-emerging, 2016.

[15] J. Lavery, "The Future Is Now: Applications Protect Themselves against Attacks," Veracode, [Online], Available: https://www.veracode.com/blog/2016/06/future-now-applications-protect-themselves-against -attacks, 2016.

[16] S. Giehl, "Device-detector," Github, [Online], Available: https://github.com/matomo-org/device-detector, 2021.

[17] H. Wen, P. Y.-R. Huang, J. Dyer, A. Archinal and J. Fagan, "Countermeasures for GPS Signal Spoofing," Proceedings of the 18th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2005), pp. 1285-1290, Long Beach, CA, 2005.

[18] C. Gonzalez, "Adaptive Standalone Secure Software," US Patent 10,521,613, B1, [Online], Available: https://patentimages.storage.googleapis.com/eb/fb/4b/82980dc0f04d32/US10521613.pdf, Dec. 2019.

[19] D. Robinson, "The Identity Crisis of International Criminal Law," Leiden Journal of International Law, vol. 21, pp. 925–963, 2008.

[20] R. Chesney, "The CIA, Covert Action and Operations in Cyberspace," Lawfare, [Online], Available: https://www.lawfareblog.com/cia-covert-action-and-operations-cyberspace, July 2020.

[21] L. T. Greenberg, S. E. Goodman and K. J. Soo, Information Warfare and International Law, National Defense University Press, 1998.

[22] M. K. Kuschner, "Legal and Practical Constraints on Information Warfare," [Online], Available: https://www.airuniversity.af.edu/Portals/10/ASPJ/journals/Chronicles/kuschner.pdf.

[23] R. S. Dewar, "The Triptych of Cyber Security: A Classification of Active Cyber Defence," Proc. of the 6th International Conference on Cyber Conflict (CyCon), DOI: 10.1109/CYCON.2014.6916392, 2014.

[24] L. Mälksoo, Russian Approaches to International Law, Oxford, ISBN-13: 978-0198808046, 2015.

[25] C. Cai, "International Law in Chinese Courts during the Rise of China," American Journal of International Law, vol. 110, no. 2, pp. 269-288, DOI:10.5305/amerjintelaw.110.2.0269, 2016.

[26] IEEE, "IEEE Standard for Software Safety Plans," IEEE Standards Association, IEEE 1228-1994, August 1994.

[27] Joint Software System Safety Committee, "Software System Safety Handbook: A Technical & Managerial Team Approach," US Department of Defense, [Online], Available: https://dl.icdst.org/pdfs/files/42fd057643931936afc1e649cee8c723.pdf, Dec. 1999.

[28] N. G. Leveson, Safeware: System Safety and Computers, Addison-Wesley, 1995.

[29] MIL-STD-882E, "Department of Defense Standard Practice: System Safety," US Department of Defense, May 2012.

[30] C. Gonzalez, "Methods and Apparatus to Provide and Manage Security for the Access to Mobile Electronic Devices," US Patent, Patent no. US 7,941,669 B2, [Online], Available: https://patentimages.storage.googleapis.com/46/58/d5/bbc2a56707980d/US7941669.pdf, March 2015.

[31] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez and E. Vazquez, "Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges," Comp. & Sec., vol. 28, pp. 18–28, 2009.

[32] D. Frincke, A. Wespi and D. Zamboni, "From Intrusion Detection to Self-protection," Comput. Netw., vol. 51, no. 5, pp. 1233-1238, [Online]. Available: https://doi.org/10.1016/j.comnet.2006.10.004, 2007.

[33] A. Nagarajan, Q. Nguyen, R. Banks and A. Sood, "Combining Intrusion Detection and Recovery for Enhancing System Dependability," Proceedings of the IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W), pp. 25-30, Hong Kong, China, 2011.

[34] N. Stakhanova, S. Basu and J. Wong, "A Taxonomy of Intrusion Response Systems," Int. J. Inf. Comput. Sec., vol. 1, no. 1, pp. 169—184, 2007.

[35] J. J. Hagg, "A Simple Introduction to Obfuscated Code," Dream.In.Code, [Online], Available: http://www.dreamincode.net/forums/topic/38102-obfuscated-code-a-simple-introduction/, Sep. 2015.

[36] A. AlSobeh, S. AlShattnawi and A. Jarrah, "WEAVESIM: A Scalable and Reusable Cloud Simulation Framework Leveraging Aspect-oriented Programming," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 02, pp. 182-201, June 2020.

ملخص البحث:

الهدف من هذا البحث وصْف منهجية عملياتية لزيادة الأمان داخل البرمجيات العسكرية الآمنةذاتية الحماية. إنّ البرمجيات ذاتية الحماية هشّة للتَّهديدات، ويعتمد ذلك في الغالب على مُستخدم تلك البرمجيات. لذا، فإن كشف المستخدم الحالي للبرمجيات ذاتية الحماية أمر شديد الأهمية.

تشمل المنهجية المقترحة على ثلاث مراحل: كشْف المستخدم، وتحليل الحالة الرّاهنة، وردود الفعْل. وتتكون مرحلة الكشف من تقدير الموقع الجغرافي، والوقت في الموقع الرّاهن، وتحديد نوع المستخدم (صديق أو عدوّ). أما مرحلة التحليل، فتشمل تحليل ما إذا كانت البرمجيات ذاتية الحماية ينبغي أن تكون في الموع الحالي، وتوقّع المواقع المستقبلية، وتقدير مستوى التهديد. وتتضمن مرحلة ردود الفعل تحديد الأفعال الفورية أو المتأخرة –إن وجدت– وتنفيذها بناءً على الحالة.

كذلك تم التّطرق الى المسائل القانونية المتعلقة بهذا الموضوع، ووصْف الإجراءات المضادّة وردود الفعل المقترحة. ويبين النّموذج التحليلي أنّ البرمجيات ذاتية الحماية التي تحتوي على خاصية كشْف المستخدم توفّر حمايةً أكبر مقارنة بمثيلاتها التي لا تحتوي على هذه الخاصية.

DECREASING THE RA COLLISION IMPACT FOR MASSIVE NB-IOT IN 5G WIRELESS NETWORKS

Bilal Rabah Al-Doori[1] and Ahmed Zurfi[2]

(Received: 29-May-2021, Revised: 17-Jul.-2021, Accepted: 2-Aug.-2021)

ABSTRACT

To satisfy the gigantic need for Internet of things (IoT) applications, the third-generation partnership project (3GPP) has revealed the narrowband IoT (NB-IoT) standard. In any case, collisions in the radio access channel of NB-IoT can be extreme due to the numerous random-access (RA) activates by a massive number of NB-IoTs and the limited available radio resources. The RA procedure is one of the MAC-layer's functions that initiates a contention-based setup to grant an uplink transmission. In this paper, the performance of a new RA procedure is investigated by introducing a modified backoff scheme to reduce the collision probability. The key mechanism of the proposed scheme is to perform an autonomous approach for determining the time for an NB-IoT to transmit in a collide environment. The proposed scheme can improve the overall throughput of the network and the NB-IoTs battery lifetime while prioritizing some QoS parameters such as favoring the NB-IoTs with heavier traffic loads. The probability of collision analysis is subjected to many operating parameters, including the backoff countdown probability, number of NB-IoTs, queue size and the contention window size. The system and link-level simulations are conducted to assess the proposed scheme with up to five thousand NB-IoTs per cell. The simulation results showed that the proposed scheme outperforms the conventional approach.

KEYWORDS

Backoff, Collision, MAC, Random access, Window size.

1. INTRODUCTION

The Internet of things (IoT) market is expected to include numerous applications with different quality of service (QoS) constraints [1]. The third-generation partnership project (3GPP) revealed the NB-IoT, known as the narrowband-IoT (NB-IoT), which is one of the classes of IoT technologies. In essence, NB-IoT is an LTE-based cellular radio access technology that was announced in Release 13. The NB-IoT supports low data rate, long battery-life time and wide-area coverage connectivity in a licensed spectrum. However, the gigantic development of sensors, healthcare smart devices and wireless identifications associated with the Internet may make extra issues relative to the complexity of NB-IoT-based frameworks and the bottleneck deployment [2]. Existing Internet infrastructure might be deficient for managing massive IoT connectivity; consequently, new web models, correspondence innovations and plan strategies ought to be created to empower the improvement of proficient IoT networks [3].

The 3GPP specifications of NB-IoT have expanded beyond Release 13, with support for diverse requirements [4]. In Release 14, extra features such as higher data rates, multicasting and authorization of the coverage enhancements were further improved. In Release 15, 5G New Radio (NR) was regulated and intended to support various requirements, such as enhancing mobile broadband, decreasing latency and connecting a massive number of devices. In Release 16 (up to date 3GPP release), new agendas are included for finalizing the NB-IoT network development, where the main objectives of the release are grant-free access, multi-user simultaneous transmission and idle-mode mobility. The aforementioned objectives are mainly embracing the RA channel procedure at the NB-IoT MAC sublayer.

For NB-IoT connectivity, the RA procedure is a contention-based mechanism in which an NB-IoT chooses the RA shared resources for in-band communications through realizes the uplink synchronization, attains an uplink grant and establishes a connection with the gNB (base station in 5G NR).

Massive NB-IoT is one of the goals in the 5G network optimization. However, collisions could occur frequently in dense NB-IoT transmissions whenever two such NB-IoTs are transmitting at the same time

[1] B. R. Al-Doori is Researcher with the Department of Electronics and Communications, University of Baghdad, Baghdad, Iraq. E-mail: bilal.rabah@coeng.uobaghdad.edu.iq

[2] A. Zurfi is Researcher with the College of Agriculture, Uni. of Kerbala, Kerbala, Iraq. Email: ahmed.jabbar@uokerbala.edu.iq

over the same RA channel, resulting in loss of one or both transmissions. Indeed, a collision occurs when the backoff counter of two NB-IoTs ends simultaneously. It is noteworthy to mention that the backoff counter is the one that is responsible for managing the RA procedure at the NB-IoTs [5].

The main challenge of massive NB-IoT network deployment is the fact that frequent access leads to unsuccessful completion of the RA procedure. For this reason, several recent contributions studied some existing IoT protocols in terms of delay and latency, such as long-range (LoRa) and Sigfox, where the latter uses the ultra-narrowband (UNB) modulation techniques [6]-[8]. In addition, a trustworthy secure transmission to develop a reliable IoT framework is introduced in [9], specifically when mission-critical applications are employed. In [10], a new effective factor is suggested for improving the throughput of the 5G networks, whereas the numerology concept of the subcarrier spacing is adjusted to tune the clustered delay channel. It is concluded that the overall throughput is enhanced with a large numerology selection. On the other hand, some other recent contributions focused on decreasing the probability of collision, where some factors are omitted. For example, the semi-persistent scheduling technique is successfully used in [11] for decreasing the latency of IoT networks. However, this scenario involves diminishing resources during the long off-period of transmission. The work in [12] introduced an exact expression of the RA channel for the NB-IoT network considering increasing the repeated preamble attempts. This seems questionable, since such an approach will lead to waste more resources and diminish the NB-IoT's battery lifetime. The work in [13] offered a partial preamble transmission mechanism, where each NB-IoT must transmit a specific fraction of the preamble sequence during the RA procedure. However, this appears to be debatable, since typically, there is no coordination between the NB-IoTs at the RA procedure phase, where the contention-based is initiated.

We remark that all the aforementioned works cannot offer any strict guarantees on packet delivery without collision when deploying a massive number of NB-IoTs, which received much less attention in the literature. Thus, we introduce an in-depth analysis of a new backoff scheme, to be called backoff-based queue length (BBQL), to improve the countdown probability of the backoff counter and thus decrease the probability of collision during the initiation of the RA procedure. Many factors are considered, such as the initial queue lengths, number of NB-IoTs, size of the contention window and the countdown probability. By default, the NB-IoTs' backoff counters are lessened deterministically in every time slot. However, the BBQL proposed scheme changes the functional behavior of the backoff counters in such a way that the NB-IoTs of the higher traffic loads have the largest probabilities to decrement their counters. Such NB-IoTs will thus statistically reach zero and attempt to transmit their head-of-line packets earlier. The proposed scheme could eventually reduce collision and enhance the NB-IoT battery lifetime by minimizing the transmission repetitions and improving the overall throughput.

2. RANDOM-ACCESS PROCEDURE

The deployments of NB-IoTs within the 5G network include the RA procedure for filling numerous needs; for example, starting access while building up a radio connection and scheduling reservations. However, uplink synchronization is the core purpose of the RA procedure for keeping up with uplink orthogonality [1]. As presented in Figure 1, the contention-based RA procedure includes the following stages [4]: (1) NB-IoT sends Msg1, the RA preamble, preceded by system information block (SIB) signaling sends by the gNB for uplink carrier frequency and the uplink channel bandwidth to be used by NB-IoT. It is noteworthy to mention that the RA procedure can also be triggered by gNB through sending the narrowband physical downlink control channel (NPDCCH); (2) the gNB sends an RA channel response (RACH) message that includes the parameters for the third step, such as the scheduling of uplink resources and the advance timing for an NB-IoT; (3) an NB-IoT uses the reserved resources for transmitting its identity to the gNB to start the process of attaching; and (4) the gNB sends preceded NPDCCH information for Msg4. This information is used for avoiding any contention due to receiving several preambles from numerous NB-IoTs in Msg1. Then, Msg4 is provided by gNB that includes the transmission parameters to be used for NB-IoT transmission, such as subcarrier offset, number of subcarriers, periodicity of NPRACH resource and starting time of NPRACH. After the completion of the four RA messages, the hybrid automatic repeat request (HARQ) is sent by NB-IoT for confirming the successful accomplishment of the RA procedure and receiving the transmission parameters.

Then, the gNB transmits NPDCCH for granting the attach procedure followed by response form NB-IoT to finalize that.

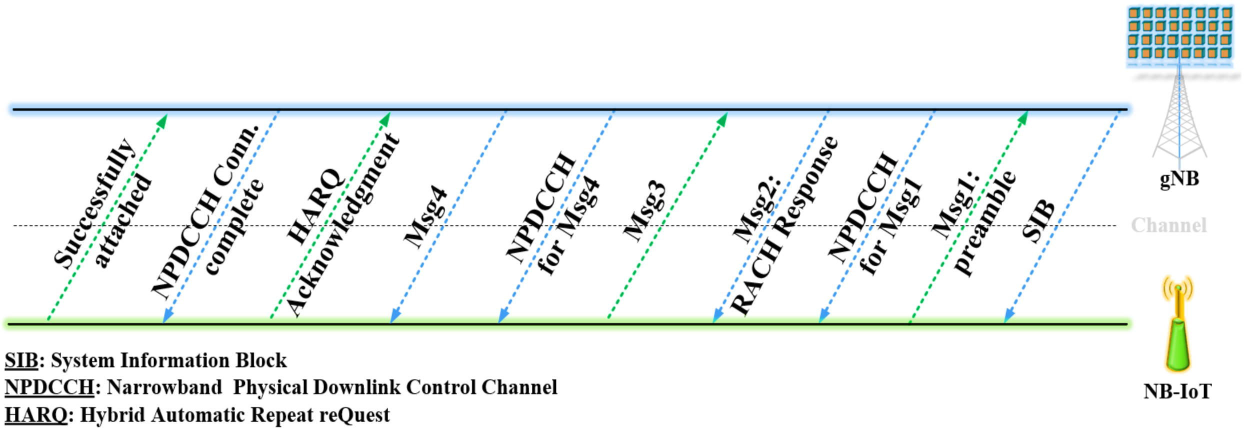


Figure 1. NB-IoT random-access procedure.

It is noteworthy to mention the following: the single-tone/multi-tone parameter is indicated by NB-IoT in Msg1 in order to let the gNB reserve the sufficient scheduling of resources for uplink transmission in Msg3. In addition, each gNB is divided into one to three coverage enhancement levels (CEL). Each CEL is used to conduct NB-IoTs with a specific range of channel quality to gNB (i.e., good or poor channel quality) and specific RA procedure settings (NPRACH resource configurations). Therefore, an NB-IoT will reattempt the RA procedure at a higher CEL with a specific preamble set, on the off chance that it does not get the RA response message after the expected number of attempts, NB-IoT proceeds to the next CEL [4]. Finally, it is important to point out that the BBQL approach is working for improving the RA procedure at Msg1.

3. The Analytical Model of the BBQL Backoff Scheme

In this model, the analysis is based on assuming that all the NB-IoTs start the backoff procedure at the same time. This allowed us to assess the NB-IoTs behavior on an equivalent basis. By default, the backoff process starts when the channel is free and ends at the moment of transmission, as shown in Figure 2. Briefly, each NB-IoT should have a specific profile for determining the transmission parameters, such as the contention window (CW) size, number of preamble repetitions and the queue size. Each NB-IoT is sensing the channel for sending. When the channel is busy, an NB-IoT is simply waiting. On the other hand, if the channel is clear, then an NB-IoT is delaying for a specific time slot and this is governed by the simulation. Then, a pre-processing step for sending the preamble is initiated by the backoff procedure. During the backoff process, as long as the channel is free, the counter discounts autonomously in every time slot. However, if an NB-IoT sensed that the channel is busy, then the counter holds till the channel is clear all over again, then the backoff counter continues the decreasing action within the CW. By the end of the last time slot, an NB-IoT starts the sending process. The abovementioned scheme presents the conventional procedure that is included within Msg1 of the 3GPP RA procedure [4].

As an alternative to the conventional step-down counter, the proposed BBQL scheme uses a weight function to determine the backoff counter value for each NB-IoT. In detail, the backoff counter could depend on the queue length of the NB-IoTs within the CW size [14]. Therefore, the BBQL scheme uses the queue length to calculate a weight function of each NB-IoT that is represented as a real number varying from zero to one. Thus, an NB-IoT that has a longer queue obtains a higher weight in the same way. Then, the countdown probability p is assigned, where the backoff counter is decreased with probability p. Accordingly, if the weight function is high (large queue length), then the probability p is high. Thus, NB-IoT countdowns quickly and wins the channel to transmit sooner.

In our analysis, we address whether a collision occurs. Therefore, we first derive the expression for measuring the backoff ending probability of NB-IoTs; i.e., determining the probability of time to transmit. Then, we conclude the expression for checking the probability that the backoff ends with a collision. In the beginning, let us use p(t) as the probability, where the backoff counter ends at a given time t. Each NB-IoT i has a backoff counter for stepping-down time slots denoted by ψ_i, such that p_i is the probability that ψ_i is decremented for each time slot, as shown in Figure 2. For each NB-IoT, the weight function q_i is used to determine the countdown probability p_i. Therefore, if there are two NB-

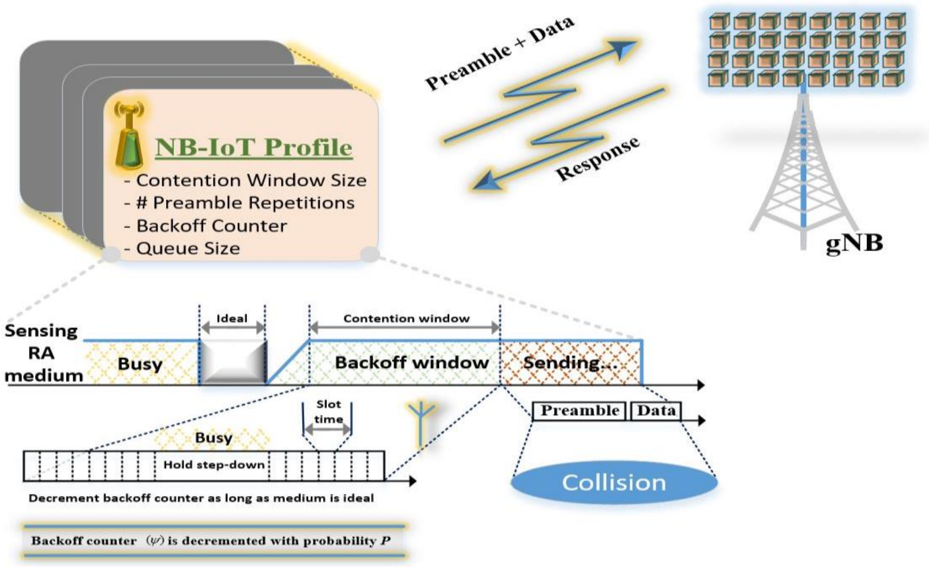


Figure 2. System model.

IoTs with different weights q_i>q_j, then p_i>p_j. For a given period of s time slots (within the range of CW size), the probability that a backoff counter is exactly decremented by k times for NB-IoT i is denoted by $\beta_i(k,s)$ and given as:

$$\beta_i(k,s) = \binom{s}{k} p_i^k (1-p_i)^{(s-k)}, \tag{1}$$

where binomial probability is used to find $\beta_i(k,s)$.

Accordingly, when the backoff counter ends at time slot *t*, the probability in which NB-IoT *i* will transmit is denoted by $\delta_i^c(t)$, where *c* is a given value of ψ_i, as shown in Figure 3.

$$\delta_i^c(t) = \begin{cases} 0 & t < c, \\ \beta_i(c-1,t-1)p_i & t \geq c. \end{cases} \tag{2}$$

For $t < c$, the backoff counter is larger than the number of time slots to decrement to zero; therefore, the probability of transmission is zero. However, for $t \geq c$, the time is long enough to reach time slot *t* and decrement to zero. It is noteworthy mentioning here that the counter must be decremented to *c*-1 times, thus in the final slot, the probability of the last decrement is p_i. Hence, by substituting (2) in (1), we get:

$$\beta_i(c-1,t-1) = \binom{t-1}{c-1} p_i^{c-1}(1-p_i)^{(t-c)}. \tag{3}$$

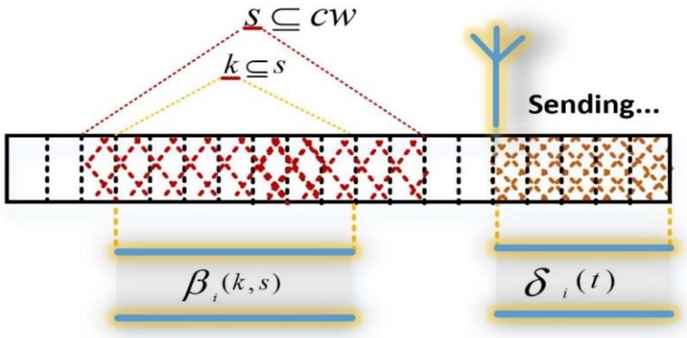


Figure 3. Probabilities' pattern.

From the above mentioned information, we can derive the total probability that the NB-IoTs could transmit at time *t* as:

$$\delta_i(t) = \sum_{c=1}^{\tau} \delta_i^c(t) P(\psi_i = c), \tag{4}$$

where τ is the CW size specified by the operator. Thus, the backoff counter is uniformly distributed on the interval $[1,\tau]$ and so we have $P(\psi_i = c) = 1/\tau$. As a result, the probability that NB-IoT *i* has no chance to transmit in any time slot until *t* is given by:

$$1-\sum_{j=1}^{t-1} \delta_i(j). \tag{5}$$

Therefore, the total probability that no NB-IoTs have the chance to transmit and the channel remains ideal until time *t* is denoted by $\gamma(t)$. Thus, we have:

$$\gamma(t)=\prod_{i\in I}\left(1-\sum_{j=1}^{t-1}\delta_i(j)\right). \tag{6}$$

During the backoff process $(0 \cdots t-1)$, all the NB-IoTs are expected to just wait and decrement their counters. However, at the end of the backoff counter, more precisely at time slot t, one of the NB-IoTs is expected to attempt transmission; i.e., δ_i conditioned from (5). Accordingly, we use $\chi_i(t)$ to refer to the probability that NB-IoT *i* transmits after the end of the backoff counter as:

$$\begin{aligned}\chi_i(t) &= \frac{\delta_i(t)}{\sum_{j=t}^{\infty}\delta_i(j)}\\ &= \frac{\delta_i(t)}{1-\sum_{j=1}^{t-1}\delta_i(j)},\end{aligned} \tag{7}$$

where the first sum in (7) leads to infinity; therefore, the second form of (7) is introduced to get a finite sum. The term $\sum_{j=t}^{\infty} \delta_i(j)$ refers to the probability of an NB-IoT *i* with index *j*, to transmit for *j* ≥ *t*. Therefore, the term $\sum_{j=1}^{t-1} \delta_i(j)$ given is more computationally tractable, since the sum is finite. It is noteworthy to mention that $\chi_i(t)$ is just a normalized form of the probability of transmission $\delta_i(j)$. In fact, $\chi_i(t)$ includes only the event space of the leftover time slots (slots at or later than *t*). Thus, we can use (7) to derive the probability that at least one NB-IoT transmits at the end of backoff (at time *t*), as:

$$1-\prod_{i\in I}(1-\chi_i(t)). \tag{8}$$

Consequently, the first goal of this analysis outlined by deriving the probability of the backoff counter ends is presented as:

$$P(t)=\gamma(t)\left(1-\prod_{i\in I}(1-\chi_i(t))\right). \tag{9}$$

The second objective of this study is to derive the probability that NB-IoT transmits first successfully and with no collision. For an NB-IoT to transmit first, we are dealing with the highest weight NB-IoT *i** and it must transmit at the end of backoff at time *t*. Therefore, we can use (7) and (8) to get the probability of NB-IoT *i** to send without collision as:

$$\chi_{i^*}(t)\left(\prod_{i\in I, i\neq i*}^{n}(1-\chi_{i^*}(t))\right). \tag{10}$$

To consider all the transmission conditions that measuring the success of the assigned values for the countdown probabilities, we derive the total probability of transmitting, no matter when the backoff ends. In other words, we consider that each value of t for each NB-IoT, where all the NB-IoTs are mute before that time slot and only NB-IoT i* sends first and without collision.

$$\sum_{t=1}^{\infty}\left(\gamma(t)\chi_{i^*}(t)\prod_{i\in I, i\neq i*}(1-\chi_{i^*}(t))\right). \tag{11}$$

Since the dense environment includes a massive number of NB-IoTs, there is a possibility that the NB-IoTs other than NB-IoT i* could transmit at the same time. Therefore, we derive the probability of transmission for NB-IoT i*, despite the consequences of the collision.

This probability can be developed by revoking the constraint that all the NB-IoTs must be silent during NB-IoT i* transmission. Thus, we cancel the product over the NB-IoTs from (11) and get:

$$\sum_{t=1}^{\infty}\gamma\,(t)\chi_{i^*}(t). \tag{12}$$

Then, we derive the collision probability for the BBQL scheme such that the backoff process ends with a collision. We use the expression in (11) and modify it to let any NB-IoT transmit, not just i*. To do this, we first find the probability of success without collision and sum over all the NB-IoTs for each possible backoff interval. Thus, the probability of success is defined as:

$$P_s=\sum_{t=1}^{\infty}\sum_{i\in I}\left(\gamma(t)\chi_i(t)\left(1-\prod_{j\in I, j\neq i}(1-\chi_j(t))\right)\right). \tag{13}$$

Finally, the complement is taken for the probability of success in (13) to obtain the collision probability; 1 - P_s.

4. System Model

For the dense NB-IoT scenario, we consider a network such that a hexagonal macrocell with gNB is adaptively working for both LTE-A and NR-5G systems. The OFDMA system is assumed to be perfectly synchronized; hence, collision happens if the same resource units are used by other NB-IoTs simultaneously. Figure 4 illustrates a high-level demonstration of the core qualities employed in the system model. Essentially, the random-access manager (RAM) functionalities include spectrum allocation and bandwidth management, where multiple carriers, subcarrier spacings, single-tone and multi-tone transmissions can be configured. On the other hand, the RAM of NB-IoTs is accessing the adaptive modulation and coding (AMC) module and the uplink packet scheduler. The AMC module namely picks the appropriate transmission parameters, such as the transport block size, modulation and coding scheme (MCS) index and number of RUs given by the chosen scheduling strategy. In addition, the CEL feature is used for enhancing the reachability as well as determining the number of repetitions required. Up to three CEL may be configured for serving devices experiencing different received power levels. According to what has been previously clarified, the scheduling paradigm is redesigned. For this purpose, the BBQL suggested scheme provides an inherent method for improving the RAM functionality of NB-IoTs' MAC layer. Most of the messages exchanging, explained in Section 2, is performed by RAM with the RA procedure. For deploying an NB-IoT, some essential factors should be considered for ensuring the functionality of the narrow and system design, such as improving system coverage, power consumption and delay tolerance.

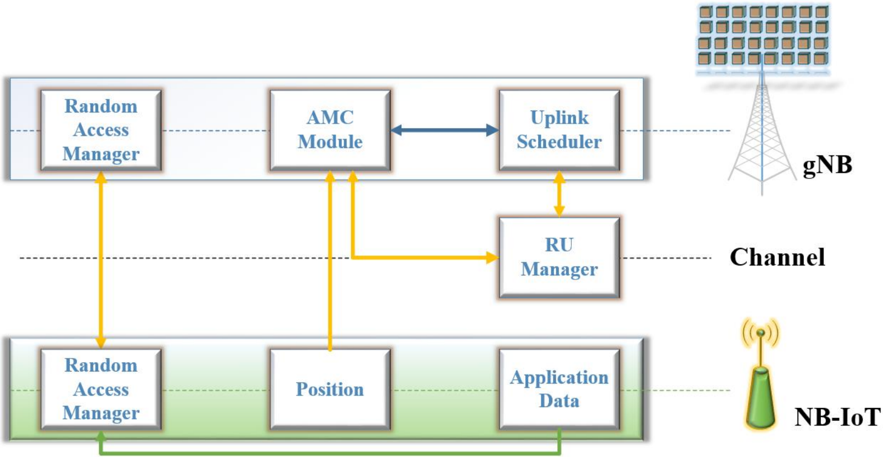


Figure 4. High-level system model.

Furthermore, each NB-IoT's profile is loaded with the operational parameters. In fact, according to the selected CEL, the NB-IoTs' MAC layer is acknowledging the operational parameters for each profile from gNB. This process is completed after a successful RA procedure. According to NR-5G specifications, these parameters primarily are numerology, number of transmission and repetitions of RACH preamble, CW size, coverage class, tones and bandwidth [5], [12].

5. Description of Simulation

The scenarios in the link and system-level simulations are implemented by using 5G-Sim [15], a framework simulator supported by Linux operating system. In the present simulation, the NB-IoTs are deployed over an area with random located positions, where constant bitrate is considered for the upload traffic flows. Table 1 summarizes the main parameters used in the simulation. In this work, the channel model is composed of path loss, shadowing, penetration loss and fast fading. The fast-fading model due to multipath depicts the small-scale parameters, such as delays, powers and the arrival and departure course on a very short time scale (within TTI interval). These parameter-based channel variations are modeled with pre-calculated traces produced according to tabulated distribution functions, as depicted in [16] and [17]. Shadowing is presented as a log-normal varying value and penetration loss is generally chosen to be constant. The path loss relies mostly upon the allocated spectrum, the position of the NB-IoT with respect to the gNB and the environment scenario. The 3D-Urban Macro-cell model in [18] is used and computed as $161.04 - 7.1\log_{10}(20) + 7.5\log_{10}(H_b) - (24.37 - 3.7(H_b/$

$H_{gNB}))^2\ 10\log_{10}(H_{gNB}) + (43.42 - 3.1\log_{10}(H_{gNB}))\ (\log_{10}(d_{3D}) - 3) + 0\log_{10}(0.001f) - (3.2\log_{10}(17.625)^2 - 4.97) - 0.6(H_{NB} - 1.5)$, where the meanings of the used symbols are as follows: d_{3D} is the three-dimensional distance between the gNB and the NB-IoT in km (counting heights in the computation), H_{gNB} is the altitude of the gNB, f is the center frequency in GHz, H_{NB} is the height of the NB-IoT and H_b is the average height of the buildings around it. Thus, the noise power is calculated as -147+ 10log(3.75)= -138.3 dBm/Hz [12], which is integrated over the bandwidth of one resource block. In addition, the 5G NR numerology of subcarriers is 3.75 kHz tone spacing; i.e., 180 kHz of the spectrum is filled with and spans over 48 subcarriers. Furthermore, the time interval between two successive transmissions by an NB-IoT is 60 seconds. On the other hand, the maximum number of retry attempts for the RACH is four. In addition, the active NB-IoTs' locations regarding the gNB are assumed to be belonging to the specific CEL with a unity probability. To obtain a statistical significance regarding the number of the deployed NB-IoTs within the cell, three categories of traces are set up and repeated for three different CW sizes. Therefore, the BBQL scheme is applied on nine random seed traces in total and the results are concluded as shown in the next section. In this way, some of the settings are fixed for all the traces such as the NB-IoTs' positions and the flows' start times and many other factors are randomized, such as the queue length and the allocated subcarriers.

Table 1. Simulation settings.

Parameter	Value	Parameter	Value
# NB-IoTs	1K, 2.5K, 5K	NB-IoT's service	Finite Buffer
CW size	256, 512, 1024	NB-IoT's application	Constant Bitrate
Bandwidth	10 MHz	Data Size	256 Byte
Tones	Single	Numerology	3.75 kHz
# Preamble TX Attempts	3	# Preamble RX Repetitions	1
Cell Radius	1 Km	Scheduling Granularity	1 Sub-frame
# NB-IoT's Carriers	5	# Coverage Class	1

6. Performance Evaluation

The performance evaluation of this work is compared with the scenario of standard configurations described in [4], (denoted as 3GPP throughout this section). Due to the purpose of this study, the performance of the overall system is conducted, where a heavier traffic load and different CW values are considered. Figure 5 shows a comparison between the 3GPP and BBQL schemes when deploying a massive NB-IoTs (1K, 2.5K, 5K) with CW size = 256. Then, the RA collision probability for each scenario is measured. In the long run of the simulation, the collision probabilities for both schemes are increasing significantly. To justify this behavior, we state the following two reasons: 1) the RA collision probability shows a non-monotonic reliance on the number of NB-IoTs.

This is because, with not many NB-IoTs deployed in the network, it is less possible that any other NB-IoTs will overwhelm i* to become the new highest-weighted node. 2) the number of NB-IoTs that are trying to complete the RA procedure and sending their data is increasing with time and since the system is set up with a heavier traffic load and low CW size, it is normal to expect such a high probability of collision. However, under these dense conditions, the BBQL scheme shows better improvements through elaborating the weight-queue metric to select the NB-IoT and tuning the backoff countdown counter. In other words, the BBQL scheme successfully ensures the probability for i^* to be the highest weighted node at the end of the backoff and remains high for all the cases. Also, the cumulative distribution function (CDF) of the waiting time in the queue is used as a metric to determine the effect of collision at the RA stage. As it can be seen in Figure 6, the 3GPP scheme with three CW size scenarios is conducted and a significant decrease in CDF is shown as CW size increases. Besides, the BBQL scheme shows an improvement as compared with 3GPP of the same CW size. This is because increasing the CW size; i.e., elaborating more time slots, will eventually increase the backoff counter numerical quantity in the system. This will give more time for another NB-IoT to complete the RA procedure and

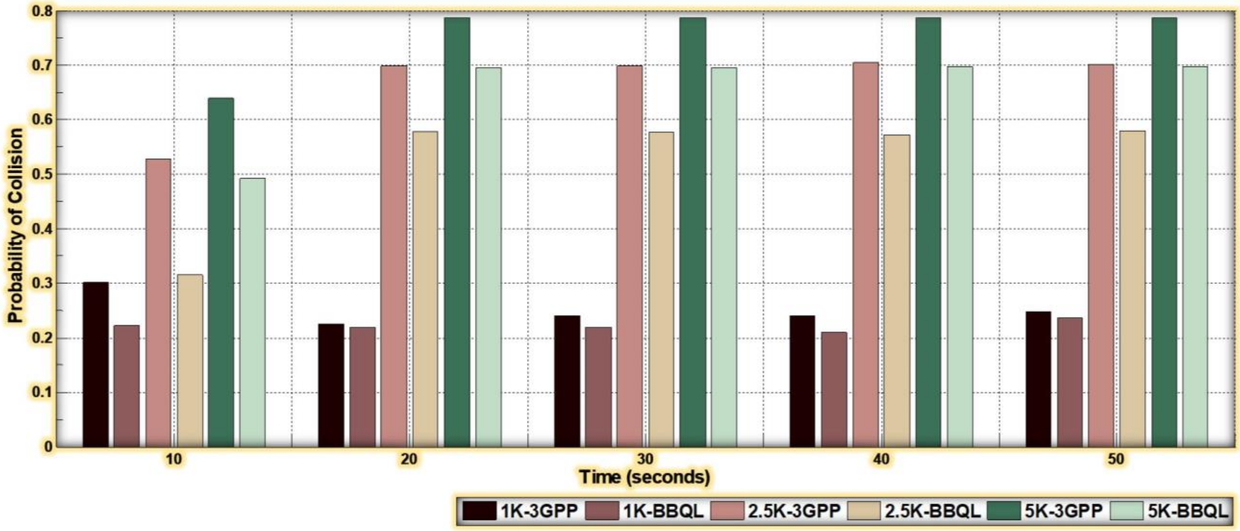
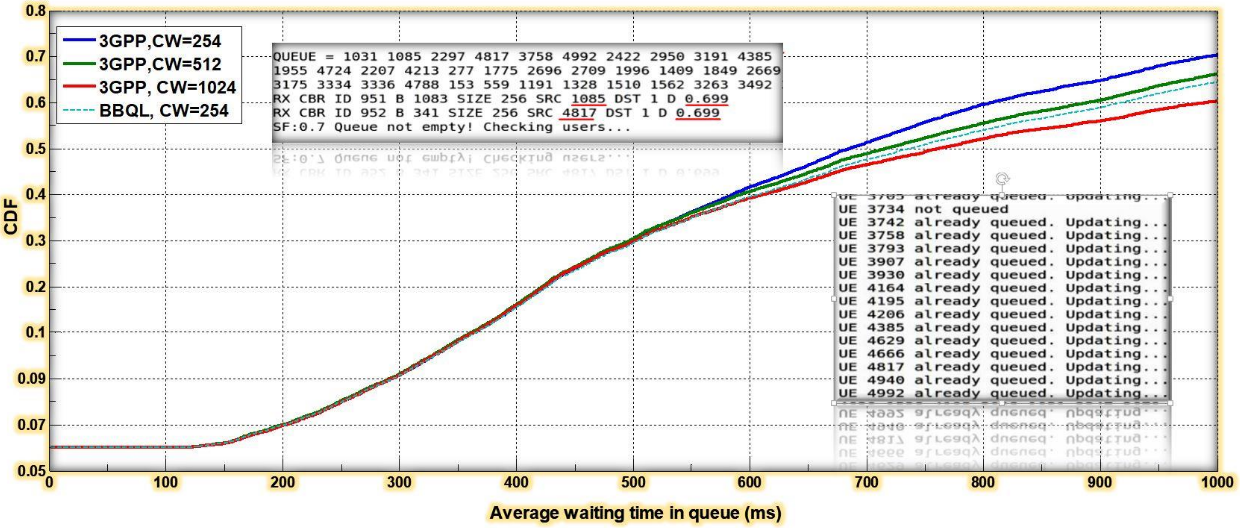


Figure 5. RA collision probability.

reduce the probability of collision. Hence, the average waiting time in the queue is minimized. However, the drawback of increasing the CW size is the fact that the NB-IoT must be in an inactive mode for a longer period to process the RA procedure and hence, diminishes the NB-IoT's battery lifetime. Moreover, Figure 6 shows snapshots of the trace captured through the simulation, where collision occurs when the NB-IoTs are trying to transmit at the same time as exemplified by the red lines.



```
QUEUE = 1031 1085 2297 4817 3758 4992 2422 2950 3191 4385
1955 4724 2207 4213 277 1775 2696 2709 1996 1409 1849 2669
3175 3334 3336 4788 153 559 1191 1328 1510 1562 3263 3492
RX CBR ID 951 B 1083 SIZE 256 SRC 1085 DST 1 D 0.699
RX CBR ID 952 B 341 SIZE 256 SRC 4817 DST 1 D 0.699
SF:0.7 Queue not empty! Checking users...
```

```
UE 3705 already queued. Updating...
UE 3734 not queued
UE 3742 already queued. Updating...
UE 3758 already queued. Updating...
UE 3793 already queued. Updating...
UE 3907 already queued. Updating...
UE 3930 already queued. Updating...
UE 4164 already queued. Updating...
UE 4195 already queued. Updating...
UE 4206 already queued. Updating...
UE 4385 already queued. Updating...
UE 4629 already queued. Updating...
UE 4666 already queued. Updating...
UE 4817 already queued. Updating...
UE 4940 already queued. Updating...
UE 4992 already queued. Updating...
```

Figure 6. CDF of average waiting time in the queue.

7. CONCLUSION

In this paper, a new backoff scheme compatible with the latest 3GPP standards, called BBQL, was introduced and examined to improve the effectiveness of the random-access procedure of massive NB-IoTs, such that the probability of collision in random access can be reduced. Theoretical analysis was conducted, where the probability of collision expressions associated with many operating parameters such as backoff countdown probability, number of NB-IoTs, queue size and contention window size is derived. The simulation results revealed that BBQL had a higher success probability that correspondingly achieves higher throughput and lower access delay than standard approaches. The performance was validated under different operating contention window sizes and the outcome confirm the efficiency of the proposed model. The one exception to this was the scenario wherein the contention window size is long enough to such an extent that the participating NB-IoTs in the network had enough backoff counter value to finish and transmit with a lower probability of collision. However, such a methodology makes an NB-IoT busy and consequently, drains the battery lifetime. Besides, the number of NB-IoTs was found to have a significant impact and as such, future efforts must utilize a reasonable number of NB-IoTs for stimulating the massive setting and participating in backoff as input to the model for ensuring practical results.

REFERENCES

[1] S. Ahmadi, 5G NR: Architecture Technology Implementation and Operation of 3GPP New Radio Standards, New York, NY, USA: Academic, 2019.

[2] S. Haq, A. Bashir and S. Sholla, "Cloud of Things: Architecture, Research Challenges, Security Threats, Mechanisms and Open Challenges," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 4, pp. 415-433, September 2020.

[3] I. Al-Joboury and E. Al-Hemiary, "Internet of Things Architecture Based Cloud For Healthcare," Iraqi Journal of Information and Communications Technology (IJICT), vol. 1, no. 1, pp. 18-26, March 2018.

[4] 3GPP, "Standardization of NB-IoT completed," [Online], available: http://www.3gpp.org/news-events/3gpp-news/1785-nb iot complete, 2016.

[5] H. Fattah, 5G LTE Narrowband Internet of Things (NB-IoT), CRC Press, 2018.

[6] F. C. de Oliveira, J. J. P. C. Rodrigues, R. A. L. Rabelo and S. Mumtaz, "Performance Delay Comparison in Random Access Procedure for NB-IoT, LoRa and SigFox IoT Protocols," Proc. of 2019 IEEE 1st Sustainable Cities Latin America Conference (SCLA), pp. 1-6, Arequipa, Peru, Aug. 2019.

[7] R. I. Ansari, H. Pervaiz, S. A. Hassan, C. Chrysostomou, M. A. Imran, S. Mumtaz and R. Tafazolli, "A New Dimension to Spectrum Management in IoT Empowered 5G Networks," IEEE Network., vol. 33, no. 4, pp. 186–193, Jul. 2019.

[8] S. Saleh et al., "5G Hairpin Bandpass Filter," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 7, no. 1, pp1-12, March 2021.

[9] Q. Tian, Y. Lin, X. Guo, J. Wen, Y. Fang, J. Rodriguez and S. Mumtaz, "New Security Mechanisms of High-reliability IoT Communication Based on Radio Frequency Fingerprint," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 1– 1, 2019.

[10] M. Almahadeen and A. Matarneh, "Performance Assessment of Throughput in a 5G System," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 3, pp. 303-316, September 2020.

[11] K.A. Nsiah, Z. Amjad, A. Sikora and B. Hilt, "Performance Evaluation of Latency for NB-LTE Networks in Industrial Automation," Proc. of 30th IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), DOI: 10.1109/PIMRC.2019.8904407, Istanbul, Turkey, Sep. 2019.

[12] N. Jiang, Y. Deng, M. Condoluci, W. Guo, A. Nallanathan and M. Dohler, "RACH Preamble Repetition in NB-IoT Network," IEEE Communications Letters, vol. 22, no. 6, pp. 1244-1247, June 2018.

[13] T. Kim, D. M. Kim, N. Pratas, P. Popovski and D. K. Sung, "An Enhanced Access Reservation Protocol with a Partial Preamble Transmission Mechanism in NB-IoT Systems," IEEE Communication Letters, vol. 21, no. 10, pp. 2270-2273, Oct. 2017.

[14] L. Kleinrock, Queueing Systems Theory, Volume I, New York: Wiley Interscience, 1975.

[15] S. Martiradonna, A. Grassi, G. Piro and G. Boggia, "5G-air-simulator: An Open-source Tool Modeling the 5G Air Interface," Computer Networks, vol. 173, DOI: 10.1016/j.comnet.2020.107151, 2020.

[16] ITU-R, Guidelines for Evaluation of Radio Interface Technologies for IMT-Advanced, Technical Report M.2135, [Online], Available: https://www.itu.int/pub/R-REP-M.2135, 2008.

[17] 3GPP, 5G; Study on Channel Model for Frequencies from 0.5 to 100 GHz, Technical Report 38901, 3rd Generation Partnership Project, [Online], Available: https://www.etsi.org/deliver/etsi_tr/138900_138999 /138901/14.00.00_60/tr_138901v140000p.pdf, 2018.

[18] ITU-R, Guidelines for Evaluation of Radio Interface Technologies for IMT-2020, Technical Report M.2412, [Online], Available: https://www.itu.int/pub/R-REP-M.2412-2017, 2017.

ملخص البحث:

لتلبية الحاجة الهائلة الى تطبيقات إنترنت الأشياء، كشف مشروع الشراكة للجيل الثالث عن معيار إنترنت الأشياء ضيّقة النّطاق. وفي كل الحالات، يمكن للتّصادمات في قناة الوصول الرّاديوية في إنترنت الأشياء ضيّقة النّطاق أن تكون كثيرة جداً بسبب العديد من حالات الوصول العشوائي نظراً للعدد الهائل من تطبيقات إنترنت الأشياء ضيّقة النّطاق ومحدودية الموارد الرّاديوية المتاحة. وإنّ عملية الوصول العشوائي هي إحدى الوظائف التي تستهل وضعاً قائماً على الاتصال من أجل الإرسال.

في هذه الورقة، يتمّ استقصاء طريقة جديدة للوصول العشوائي تهدف الى التّقليل من احتمالية التّصادم. وتتمثل الآلية الخاصّة بالطريقة المقترحة في أداء نظام مستقل لتحديد الزمن اللازم لإنترنت الأشياء ضيّقة النّطاق لتقوم بالإرسال في بيئة تمتلىء بالتصادمات. ويمكن للطريقة المقترحة أن تحسّن العبور الإجمالي للشّبكة وعُمر بطارية إنترنت الأشياء ضيّقة النّطاق من خلال تحديد الأولويات؛ كأن يتمّ تفضيل إنترنت الأشياء ضيّقة النّطاق ذات الحِمل المروري الأعلى. والجدير بالذّكر أن احتمالية التّصادم تعتمد على عدّة متغيّرات تشغيلية؛ منها عدد وحدات إنترنت الأشياء ضيّقة النّطاق، وحجم الدّوْر، وحجم نافذة الاحتواء...الخ.

وقد أجريت محاكاة لتقييم النّظام المقترح لعددٍ من وحدات إنترنت الأشياء ضيّقة النّطاق وصل الى خمسة آلاف لكلّ خليّة. وبيّنت نتائج المحاكاة أنّ النّظام المقترح تفوّق من حيث الفعاليّة في التقليل من تصادم الوصول العشوائي في الشّبكات اللاسلكية القائمة على إنترنت الأشياء ضيّقة النّطاق، مقارنة بالطّرق التّقليدية.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 03, September 2021.

A New Adapted Canny Filter for Edge Detection in Range Images

Mohamed Cheribet[1] and Smaine Mazouzi[2]

(Received: 25-May-2021, Revised: 20-Jul.-2021, Accepted: 11-Aug.-2021)

Abstract

Image segmentation remains as one of the most important tasks for image analysis and understanding. It deals with raw images in order to prepare them to be usable in automatic high-level processes, such as classification or information retrieval. We present in this paper a new adapted edge detector for range images. Its principle is inspired from the Canny detector, so the inherent features of range images will be considered. Usually, Canny detector is used with greyscale or color images, where its direct application with depths does not provide satisfactory results. From the raw image, containing measured depths, a relief image that consists of an image of normal vectors to the local surfaces is computed. So, angles between neighboring vectors are used to compute an angle-based gradient. The latter is integrated in the Canny algorithm, so an edge map is produced for the range image. Real images from the ABW database were used in experimentation, where the proposed new detector has outperformed the original Canny one by a ratio of 18 %.

Keywords

Segmentation, Edge detection, Canny detector, Range images.

1. Introduction

Contrary to 2-dimentionnal (2D) color and grayscale images, range images allow non-ambiguous representation of observed scenes. They are so preferred in robotic vision, where the image analysis should be reliable and must provide a concise representation of the objects that they contain. In the past decades, it was not easy to produce depth images, given that the range devices were rare and mostly used within research laboratories and range image datasets were also few [1]. Since the commercialization of the Kinect device, range images have found an unceasing interest and datasets were multiplied during the last decade. However, the produced range images are highly distorted and affected by a high level of noise, which makes them hard to process according to conventional image analysis representations and methods.

Furthermore, image segmentation remains the most important and critical task for image analysis and recognition. It consists in the partition of an image in its intelligible parts that depend on the application to which it is dedicated. Therefore, choosing the well representation method and the well detection one is very important to ensure the expected efficiency of the whole image analysis system. The representation of raw data in images takes into account the nature of the features to be extracted and how they will be computed. Image segmentation aims at extracting from the raw image, according to the considered features, pixels of interest that can be used to delimitate parts of interest within the image, which are called regions. However, several artifacts make hard image segmentation, where every method should model and deal with such artifacts. For both main kinds of image artifacts, caused respectively by uncertainties and inaccuracies that are produced during image acquisition, several authors have proposed different methods for edge detection, based on stochastic processing [2] or fuzzy processing [3], where they have dealt with such kinds of artifacts. In [4], Mario and Morabito introduced and evaluated a fuzzy edge detector based on fuzzy divergence for edge detection, where a fuzzy entropy minimization was applied for threshold's selection. In another work, introduced by Fang et al. [5], the authors, aiming to deal with noise and intensity non-uniformity, used a fuzzy energy functional of both edges and regions. Region energy is composed of a hybrid fuzzy region term and a local fuzzy region term. The edge energy allows to maintain the appearance of smoothness by regularizing the pseudo-level set function (LSF) during the curve evolution.

[1] M. Cheribet is with Department of Computer Sciences, Badji Mokhtar University, Annaba, Algeria. Emails: mcheribet@gmail.com, m.cheribet@univ-skikda.dz

[2] S. Mazouzi is with Department of Computer Sciences, University of 20 Août 1955, Skikda, Algeria. Emails: mazouzi_smaine@yahoo.fr, s.mazouzi@univ-skikda.dz

Like for 2D images, range image segmentation methods are numerous. They differ according the feature representation method or according to the detection one. Therefore, range image segmentation methods can be split into three groups; namely: edge-based, region-based and classification- and clustering-based. Edge-based methods proceed by detecting discontinuities in the image data, using several techniques, mostly geometrical [6]-[7]. In range images, the proposed methods in the literature aim at extracting the borders of the regions in order to delimitate the latter [7]. Edge-based methods are well known for their reduced processing time. However, they suffer from lack of expressivity of the parts of images in terms of regions. Moreover, edge-detectors, such as Canny detector, are highly sensitive to noise and distortion [8], which makes them less-suited for range images.

Region-based methods rely on some homogeneity criteria to extract regions, based on the fact that those regions are composed of homogeneous pixels, according to the considered criterion. These methods are robust against noise, which makes them the most used for range images [9]-[10], [11]-[12]. Unfortunately, region-based methods are time-consuming and in most of the cases they depend on the seed from where the region starts to be extracted. Such a fact does not encourage their utilization with real-time applications, such as with robots and drones. Classification-based methods rely mostly on machine-learning techniques to label the pixels of the image according to a semantic criterion [13]-[14]. An active field of research was born from combining depth and color in RGBD (Red, Green, Blue, Depth) images, called object-based image segmentation, where machine-learning techniques are massively applied. Classification methods require learning, which cannot be always performed, due to lack of training data. Furthermore, extracted regions according these methods are not contiguous, which requires further processing.

In range images, noise processing is a dilemma. If it is strong, by using wide filters, or by applying the filter several times, some edges within the image will be smoothed and erased, so they will not be detected during segmentation. Otherwise, if the smoothing is under-performed, in order not to erase edges, the remaining noise within the image disturbs the detection and results in discontinuous and dislocated edges with wide regions of noise [15]. Fixing the level of smoothing is a challenge, in particular with acquisition constraints such as in robot vision systems. For instance, with range images, Canny detector produces mediocre results independently of the level of noise smoothing or other image enhancement techniques. Such an issue has motivated us to propose a new data representation in range images and reformulate some Canny steps, so that the resulting edge detector will be well-suited for range images.

So, in this paper, we are inspired from Canny detector for range images in order to propose a novel detector, well-suited for range images. Contrary to the conventional use of Canny, where the gradient vector at every pixel of the image is calculated based of the raw image data (color or gray level), we use angles formed by adjacent normal vectors to the surface to calculate both the magnitude and the direction of the gradient. Such calculation of the gradient allows to have high magnitudes on the pixels that are on the edges separating regions. However, magnitudes of the gradient are low on the pixels within the homogenous regions. In range images, adjacent pixels that belong to highly inclined surfaces have distant range values which do not allow to use the raw range as data to compute the gradient. Nevertheless, the orientations of the normal vectors are close within this type of surfaces, allowing to consider them for edge detection or region extraction.

The remainder of the paper is organized as follows: Section 2 introduces some well-referenced works having dealt with range image segmentation. Section 3 is devoted to the proposed method, in which we show how the surface-based image is computed and how the new gradient vector is computed; that is what consists our adaptation of the Canny detector for range images. Experimentation, results and a discussion are introduced in Section 4. Finally, Section 5 overviews the contribution of the introduced work and underlines its potential perspectives.

2. RELATED WORK

Range image segmentation depends strongly on the objects that can appear in images. According to such a fact, range image segmentation methods can be split into four families, depending on the used homogeneity criterion. These are: plane-based, curve-based, algebraic surface-based and continuity C1-based methods. In plane-based methods, suited geometrical criteria are used, such as plane equation and plane orientation. The considered criteria are then used to perform a region extraction or eventually an edge detection, according to the adopted method. It is also necessary to deal with surfaces that have the same orientation but belong to different parallel surfaces. The method of Panrin and Medioni [15] belongs to this family, where the authors used a split and merge technique with the surface normal vector as image feature. Homogeneity is based on the angles between normal vectors and is used for region fusion as post-processing. A split and merge technique based on the gradient was also used in both Taylor [16] and Bhavsar [17].

In curve-based methods, authors use thresholding methods on the mean and the Gaussian curvatures. However, the estimation of curvatures is problematic in this category of methods. Indeed, the noise due to measurement errors and depth discontinuities does not allow a good estimation. Therefore, to obtain a good estimations of curvature, it is necessary to deal with noise and eliminate the disturbances due to discontinuities by an appropriate treatment. Besl and Jain [18] used this method to set initial seeds for growing regions. Detecting discontinuities for a curve-based segmentation was proposed by Yokoya and Levine [19]. In another work, Kasvand [20] pre-processed pixels belonging to local neighborhood in order to mitigate effects due to discontinuities.

Segmentation into algebraic surfaces, which are not strictly plane, concerns two categories of 2.5-dimensional (2.5D) and 3-dimensional (3D) surfaces. 2.5D surfaces that match polynomial functions of two variables can be applied only for scalar images. For 3D surfaces, which correspond to quadric or super-quadric surfaces, they require more complex processing than 2.5D ones. Gupta and Bajcsy [21]-[22] presented a segmentation method which produces as a result the description of the range image by super-quadrics such as ellipsoids. Jiang and Bunke [23] used the growing region method under 2.5D constraints of plane approximation. This special growing method is based on line segments, where regions are formed by lists of segments. So, growing a region is executed segment after segment and linear segments are delimited according to a profile division (by column or by row) of the processed image.

In continuity C1-based methods, segmentation is based on a given criterion defining the homogeneity of a surface, which is called C1-continuity. Two principles can be distinguished:

1) Merging of segments resulting from a more constrained segmentation [18].
2) The growing of detected border points is performed to form closed borders [24].

Recently, Deep Neural Networks have been widely used for image classification and object detection and recognition. However, in several cases, post-processing based on extracted edges is required to accomplish the recognition task, such as in the work introduced by Y. Wong et al. [25], where after having recognized Racing Bib Numbers (RBNs) by using YOLOv3, they proceeded by non-maxima suppression in order to predict a single bounding box for each target object.

After reviewing the different methods, we can conclude that most of them are based on former implementations that were used with color or grayscale images. Following the same roadmap, we present in the remainder of this paper a novel edge algorithm, inspired from Canny detector, aiming at accurately detect edges in range images.

3. Adapted Edge Detection for Range Images

In this section, we present our method for edge detection in range images and introduce the necessary basic algorithms to implement it. In an earlier work [26], we have introduced an overview of the proposed detector. So, in the current paper, we extend the work by introducing the full method and its extensive experimentation, as well as a result comparison with those of other 2D-image dedicated detectors.

3.1 Original Canny Detector

The Canny edge detector [27] is an edge detection that uses a multi-stage algorithm to detect a wide range of edges in 2D images. Filter-based edge detection consists in locating high impulsion responses of the used filter. The approach used by Canny is based on the quality of criteria required for an optimal edge detector. Canny's algorithm, which uses a gradient calculation operator, such as Sobel, is designed to be optimal according to three criteria:

- *Good detection:* the algorithm should mark the whole real edges in the image as much as possible.

- *Good localization:* edges marked should be as close as possible to the real edges.
- *Minimal response:* a given edge in the image should only be marked once and where possible image noise should not create false edges.

To satisfy these requirements, Canny used the calculus of variations, a technique which finds the function which optimizes a given functional. The optimal function in Canny detector is described by the sum of four exponential terms, but can be approximated by the first derivative of a Gaussian.

Canny detector allows to detect edges that correspond to significant and quick variations in image data. However, it can ignore slow variations resulting in losing edges in the final edge map. Indeed, on the one hand, if the used threshold of the gradient norm is low, the resulting edge map contains all the true edges, but with a high amount of noise. On the other hand, if the threshold is high, noise is low, but several true edges could be ignored.

Before introducing the proposed detector for range images, we review the steps that the original Canny detector follows. They are as follows:

1. *Noise reduction*: The first step is to reduce noise before detecting edges. The 2D filter uses a Gaussian [28] that is expressed as follows:

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}} \qquad (1)$$

2. *Gradient calculation:* After noise reduction, the next step is to apply a gradient calculation, which returns intensities of edges. The gradient operator, (Roberts, Prewitt, Sobel) [29] for example, returns an estimation of the first derivative in the horizontal direction (G_x) and the vertical direction (G_y). The gradient norm *N* and its direction *θ* are then calculated at every pixel in the image:

$$N(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \qquad (2)$$

$$\theta = tan^{-1}\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \qquad (3)$$

   The edge direction angle is rounded to one of four angles, representing horizontal, vertical and the two diagonals (0, 90, 45 and 135 degrees).

   First and using the gradient norm *N*, a thresholding is performed aiming at keeping only true candidates of edges. Such an operation results in a first edge map, represented by a 2D array *EdgeMap:*

$$EdgeMap(x,y) = \begin{cases} 1 & ifN(x,y) > threshold \\ 0 & else \end{cases} \qquad (4)$$

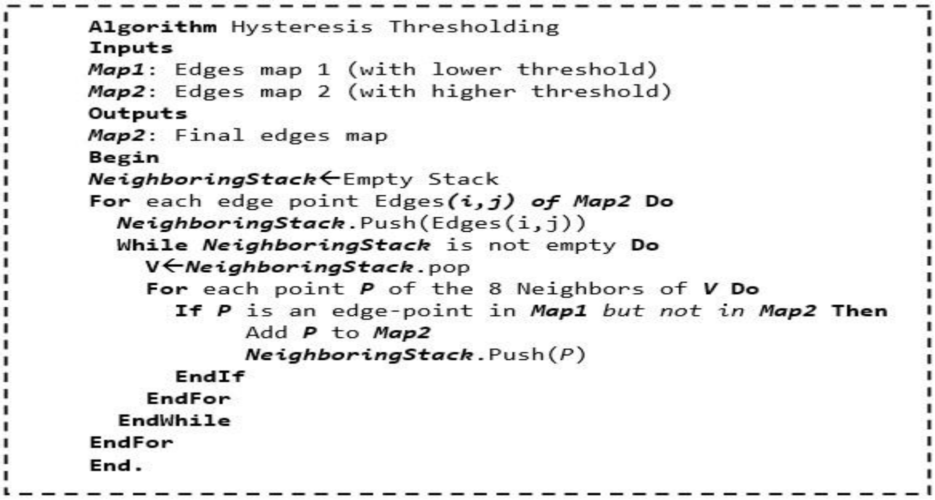
3. *Non-maximum suppression*: The gradient map records an intensity at each pixel in the image. A high intensity indicates a high probability that an edge is present at that pixel. However, this intensity is not enough to decide whether a pixel corresponds to an edge or not. Only pixels corresponding to local maxima are considered edge pixels and are kept for the next step of the algorithm. A local maximum is located at the pixel where the derivative of the gradient norm is null.

4. *Double threshold:* Two different thresholds are used and two edge maps are calculated, one with a low threshold and the other with a high threshold. Low threshold is used to identify the non-relevant pixels (intensity lower than the low threshold). So, it can detect the majority of edges and even noise. High threshold is used to identify the strong pixels (intensity higher than the high threshold). So, it only finds true and noise-free edges. However, it can miss some true edges. Moreover, non-maxima suppression should be applied to both edge maps.

5. *Hysteresis thresholding:* The differentiation of the edges on the generated map is done by hysteresis thresholding. This requires two thresholds; a high threshold and a low one, which will be compared to the intensity of the gradient of each pixel. If the intensity of the gradient of each point is less than the low threshold, the pixel is rejected. If it is greater than the high threshold, the pixel is accepted as forming an edge. If it is between low threshold and high threshold, the pixel is accepted if it is connected to an already accepted pixel. Therefore, the two obtained edge maps are merged where the second, obtained with the high threshold, is preferred, if needed. So,

we begin with the edges from the second map, that are mostly noise-free and then complete with edges from the first map.

The pseudo-code below shows a recursive version of thresholding by hysteresis:



```
Algorithm Hysteresis Thresholding
Inputs
Map1: Edges map 1 (with lower threshold)
Map2: Edges map 2 (with higher threshold)
Outputs
Map2: Final edges map
Begin
NeighboringStack←Empty Stack
For each edge point Edges(i,j) of Map2 Do
  NeighboringStack.Push(Edges(i,j))
  While NeighboringStack is not empty Do
    V←NeighboringStack.pop
    For each point P of the 8 Neighbors of V Do
      If P is an edge-point in Map1 but not in Map2 Then
           Add P to Map2
           NeighboringStack.Push(P)
      EndIf
    EndFor
  EndWhile
EndFor
End.
```

3.2 Proposed Edge Detector for Range Images

Range images represent the distances to the surface points from an observer, usually situated at the range camera. Therefore, the calculation of the gradient vector by first derivation of the raw image function is not suited for range images. This is due to the discontinuities that exist between different objects or between surfaces of the same object.

Unlike 2D images, where the intensity values of the neighboring pixels are enough to calculate the gradient, in range images pixels in a local 2D neighborhood do not necessarily represent true 3D neighborhood, resulting in what is called 3D bias (Figure 1).

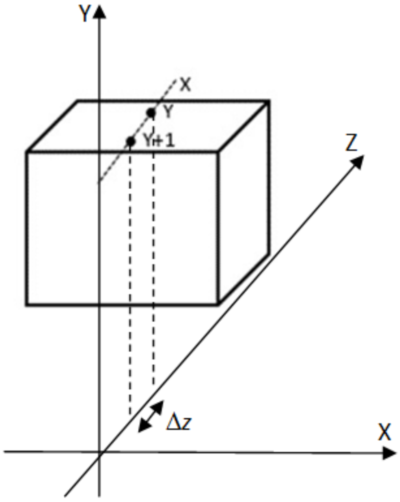


Figure 1. 3D bias in range images: (x,y) and $(x,y+1)$ are close in 2D, but distant according Z ($\Delta z>>1$).

The discontinuity of the data constitutes a characteristic of range images compared to color or grayscale images. Applying the original Canny filter to range images results in both high amount of outliers and lost edges. In the image in Figure 1, the distance $\Delta z = |I(y + 1, x) - I(y, x)|$ is high when the surface on which the two points are situated, is oriented closely orthogonal to the observation direction. Indeed, pixels (x, y) and $(x, y + 1)$ are neighbors in the plane (X, Y), but they are far from each other in space. In 2D color or grayscale images, this situation does not happen, because the differences are uniform regardless of the surface orientation.

Therefore, a well-appropriated set of features should be synthesized and then applied for gradient computation in range images. For our proposed detector, we compute a new surface-based image, by fitting the plane equation at every pixel of the raw image and then we consider the parameters of the obtained equation as the set of features for gradient vector calculation. Therefore, we can summarize our contribution in two points:

1) Generating a new surface-based image that allows to quantify variations of normal vectors to surfaces. Such variations are the basis for the calculation of the new gradient vector.
2) Adapting Sobel filter to use angles from the surface image rather than raw data from the range image (as in 2D images).

The calculation of the plane equation at a given pixel is performed either by cross-product considering two pairs of vectors, defined by three adjacent pixels, or by a multi-regression technique, as in several works in the literature [1]. Nevertheless, it has been noticed that most of the reviewed methods compute approximations of the fitted plane for an entire surface, which makes the calculation of the equation less accurate. To overcome such issue and as we focused only on how the edge can be detected locally, we were interested in how to represent the surface locally and to be able to calculate an appropriate gradient vector allowing edge detection according to the Canny detector steps. So, we will be only restricted to a limited neighborhood around each pixel, aiming at checking whether it is an edge pixel or not.

Moreover, range images are highly noisy by nature and require a smoothing operation to reduce noise without erasing edges. This should be done in the preprocessing step. In addition, the calculation of the gradient vector involves the use of the features obtained from a set of pixels belonging to a local neighborhood. Thus, an error occurring at a given pixel can be recovered by taking into account the features of the neighboring pixels.

According to our adaptation, the Canny steps for range images are modified only for the method to calculate the gradient vector and to synthesize suited features that must be used. The other steps remain unchanged (Figure 2). Indeed, the latter steps do not depend on what features have been used.

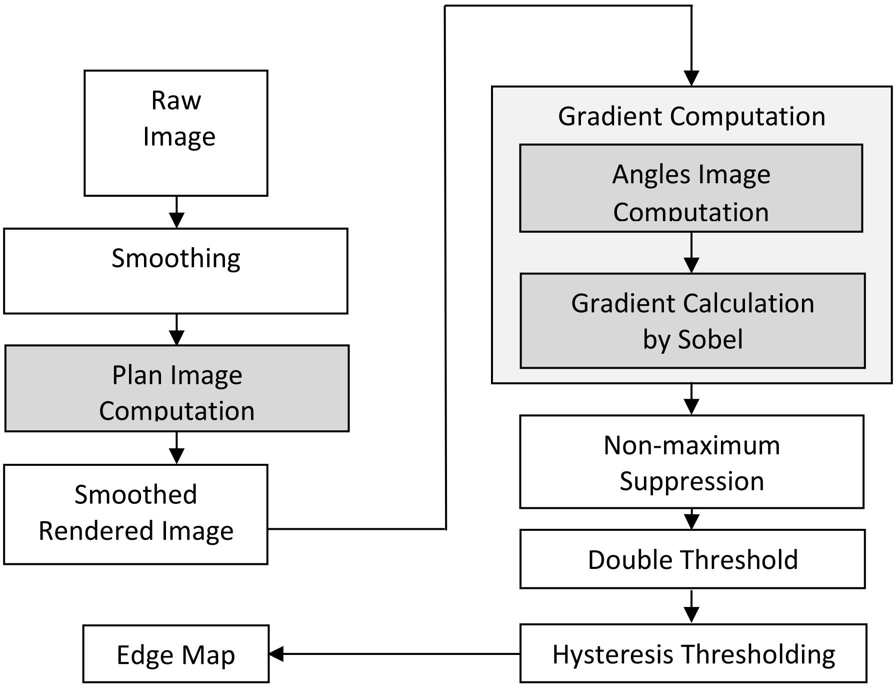


Figure 2. Flowchart of the overall proposed detector. The calculation of the smoothed image is moved outside the Canny algorithm, so the surface-based image can be computed. Steps in gray are those where adaptation is performed.

In the next part, we will present our new proposed method for calculating the gradient. This method has the advantages of being fast and remaining well suited to the peculiarity of range images that was introduced above.

3.3 Angle-based Edge Detection

Our proposed method consists in calculating a gradient vector, which measures the level of variation in surface orientation. Therefore, the estimation of the gradient vector at a given pixel (x, y), requires the calculation of plane equations according to the directions defined by the neighboring pixels. The norm and the direction of the gradient are calculated by using the angle between the obtained normal vectors at the pixels in the neighborhood. However, for a given pixel, we calculate the equations of the planes defined according to the eight possible subsets of 4 pixels in the neighborhood, as shown in Figure 3.

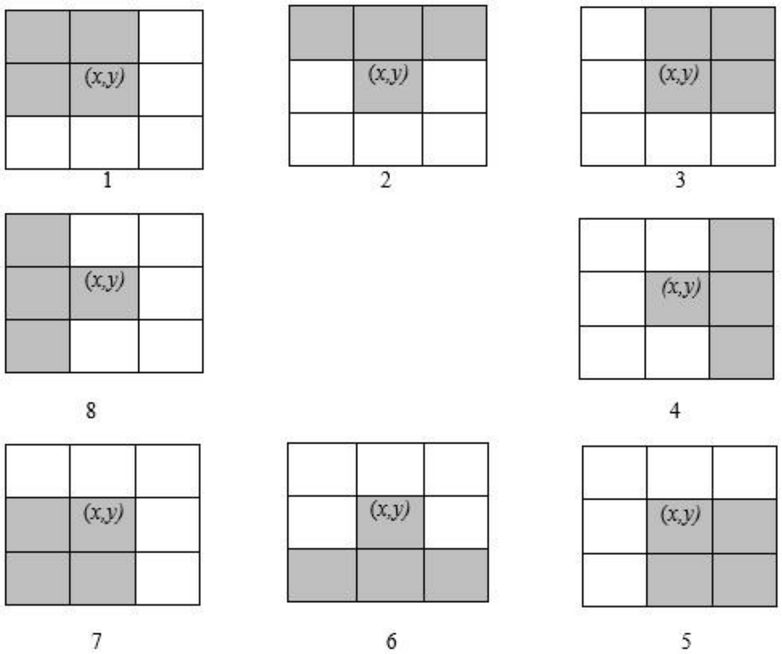


Figure 3. Used subsets of pixels to fit the plane equations for one pixel (x,y). All combinations of the central pixel with its 3 neighbors are considered.

To approximate the norm of the gradient that allows us to test whether a given pixel can be an edge pixel or not, we compute the equations of the planes in the neighborhood. For each plane, we use the central pixel and three of its neighbors:

Given a pixel (x, y), we use the set of neighboring pixels $\{(x + \Delta x, y + \Delta y); \Delta x, \Delta y = -1 .. 1\}$ to define eight planes. Table 1 below shows the list of pixels involved in the calculation of each equation.

Table 1. Sets of pixels involved in the calculation of the eight plane equations.

1	*(x, y) ; (x-1, y) ; (x-1, y-1) ; (x, y-1)*
2	*(x, y) ; (x-1, y-1) ; (x, y-1) ; (x+1, y-1)*
3	*(x, y) ; (x, y-1) ; (x+1, y-1) ; (x+1, y)*
4	*(x, y) ; (x+1, y-1) ; (x+1, y) ; (x+1, y+1)*
5	*(x, y) ; (x+1, y) ; (x+1, y+1) ; (x, y+1)*
6	*(x, y) ; (x+1, y+1) ; (x, y+1) ; (x-1, y+1)*
7	*(x, y) ; (x, y+1) ; (x-1, y+1) ; (x-1, y)*
8	*(x, y) ; (x-1, y+1) ; (x-1, y) ; (x-1, y-1)*

By considering the set S of the involved pixels, the equation of the plane $ax + by + cz + d = 0$ is calculated by linear multi-regression. As all the surfaces are visible and oriented towards the observer, the parameter c is necessarily non-zero. Therefore, the equation of the plane can be expressed by $z = \alpha x + \beta y + \gamma$, where α, β and γ are obtained after minimizing the objective function $\Phi(\alpha, \beta, \gamma)$ that expresses the least squares as follows [30]:

$$\Phi(\alpha,\beta,\gamma)=\sum_{p\in S}\left(\alpha x_p + \beta y_p + \gamma - \mathrm{I}(x_p, y_p)\right)^2 \qquad (5)$$

In the next step, we calculate the angle $\phi_{x,y}$ formed by the two normal vectors for each pair of pixels $\{(x, y) ; (x + \Delta x, y + \Delta y)\}$. Then, we calculate the gradient vector by introducing a modified Sobel operator, as follows:

According to the plane equation $z = \alpha x + \beta y + \gamma$, the normal vector is $\vec{V}_1(\alpha, \beta, -1)$. The angle $\phi_{x,y}$ defined between the normal vector and the vertical plan *(XY)*, expressed by the vector $\vec{V}_2(0,0,1)$, is calculated as follows:

$$Acos\ \phi_{x,y} = \frac{\vec{V}_1 * \vec{V}_2}{||\vec{V}_1|| * ||\vec{V}_2||} \tag{6}$$

The set of angles $\{\phi_{x,y}\}$formed by 3D normal vectors to the surfaces and the vertical plane *(XY)* are considered to calculate the gradient vector, instead of the pixel values $I(x, y)$ (Figure 4).

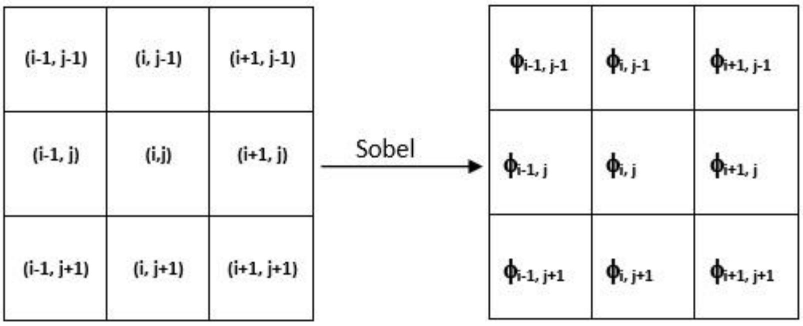


Figure 4. New features for gradient calculation using a modified angle-based Sobel detector. Raw ranges are transformed into angles.

The new gradient vector components (G_x^ϕ, G_y^ϕ) are obtained by the Sobel horizontal and vertical operators. The angle $\theta = Atan\ (\frac{G_y^\phi}{G_x^\phi})$ defines the gradient vector direction.

The obtained norms and angles of the gradient vector, so computed at every pixel of the range image, are used as inputs for the further steps of the Canny algorithm, which remain unchanged for our proposed detector.

4. EXPERIMENTATION

For the experimentation of our method, we used the ABW database that was dedicated to range image segmentation [1]. As for the original Canny detector, Gaussian smoothing is performed in order to reduce the noise in the image. After several tests by varying the parameter σ of the Gaussian filter in the range [0.1 .. 1.0], we have set σ to 0.8, for which edges in the tested images were correctly detected. Using Gaussian smoothing instead of other noise filtering techniques is due to the nature of noise within range images. Indeed, in such images, noise in mainly due to distortions in image data rather than to impulsive outliers, where median filters could be well-suited.

4.1 Qualitative Evaluation

As in our earlier work, published in [26], we begin by introducing some samples of range images from the ABW database, with their edge detection results using the original Canny detector and the proposed one. Such a visual presentation of the results allows the reader to have an idea on the advanced visual quality of edge detection by using our proposed detector. Figure 5.a shows a range image from the ABW database. It is displayed according to its raw depth data. The black regions are the shadows where the laser ray did not reach. At each pixel, the depth is represented by a gray level ranging from 0 to 255. In Figure 5.b, which represents a rendered image of the raw data, using a simple rendering algorithm with simulation of a light source, we can notice high level of noise as distortions of the plane surfaces, due to less precise range measurements.

The result of the application of the original Canny detector, using the raw image data, is shown in Figure 6.a for the high threshold and in Figure 6.b for the low threshold. The first image (Figure 6.a) shows clearly an under-detection of edges, where we can notice that several edges have not been detected, in particular those which form the borders between the surfaces of the same object. For such edges, there is a continuum in range data, so the gradient vector will be continuous at these points. On the other hand, we can clearly notice an over-detection of edges in the second image (Figure 6.b). Indeed, many

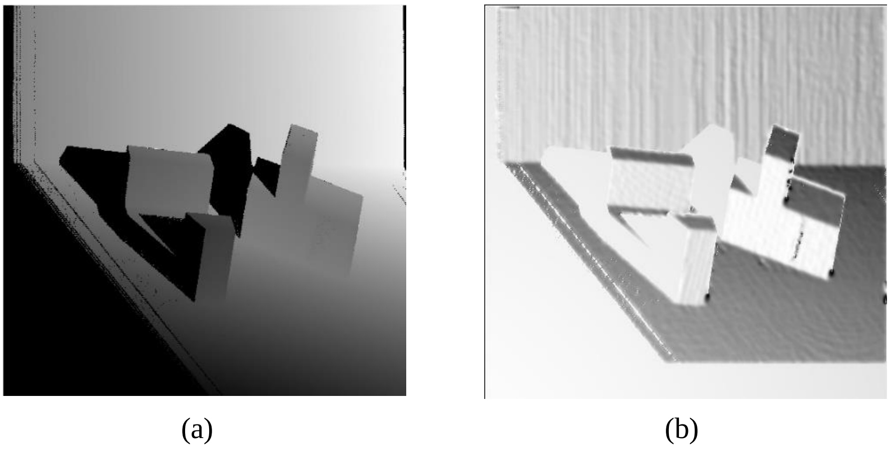


Figure 5. A sample image from the ABW dataset, (a) Range image, (b) Rendered image showing the nature of noise in such images.

pixels inside different planar surfaces were detected as edge pixels, when they are not. Otherwise, most of the edges defining borders between surfaces have been detected. However, the pixels of some surfaces highly inclined were detected all as edge pixels. The later stages of Canny's algorithm; namely non-maximum suppression and thresholding by hysteresis, cannot improve such results in the first image and generate a lot of outliers in the second.

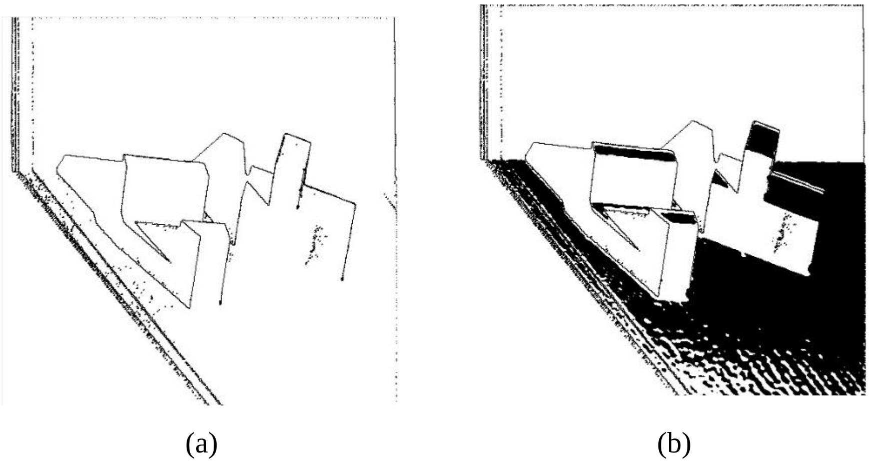


Figure 6. Edge detection in the image abw.test.3 with the original Canny detector that uses the raw data: (a) Detected edges with a higher gradient threshold (0.2), (b) Detected edges with a lower gradient threshold (0.12). An under-detection is noticed in (a), where an over-detection is noticed in (b).

After applying our proposed detector, we were able to obtain the results presented in Figure 7. The detection results obtained before non-maxima removal and hysteresis thresholding steps are shown in Figure 7.a. We can notice, unlike the original Canny detector, that our adapted detector which uses a modified gradient, based on the variation of the normal angles, allowed to produce an edge map, usable for the further Canny steps where an adequate post-processing can be performed.

The final detection result is shown in Figure 7.b. This result is obtained after having applied non-maxima suppression and hysteresis thresholding on the image gradient of Figure 7.a. The best value for the lower threshold according to the proposed detector was 0.12 radian (6.86°) and that of the higher threshold was 0.20 radian (11.46 °). Such values were obtained by varying the two thresholds in the range [0.05 .. 0.25], by 0.05 as step. We have considered the values for which the detector result was the best for the set of ABW training images. We conclude that the visual results of edge detection were satisfactory for the whole set of test images. We can also say that the edges were correctly detected, including those belonging to intra-object boundaries. The latter are difficult to detect as it has been reported in all the works having dealt with range images segmentation.

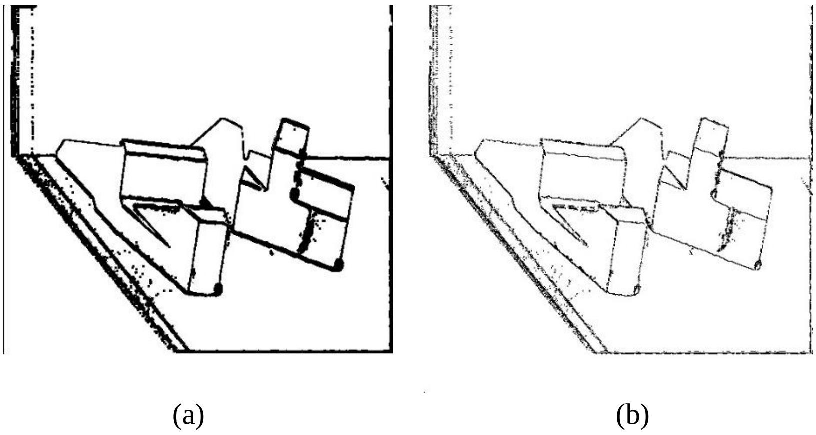


Figure 7. Edge detection in the image abw.test.3 with the proposed detector: (a) Detected edges before non-maxima suppression and hysteresis, (b) Detected edges after non-maxima suppression and hysteresis (final result).

4.2 Quantitative Evaluation

As far as we know, there is no native method for edge detection in range images; all the proposed methods we found in the literature are region-based or Machine Learning-based [31]. Moreover, the pure range images such as those of ABW, or those of other RGBD datasets that provide depth modality, such as OSD (Object Segmentation Database) [32], do not provide a ground-truth edge detection. Therefore, it will be hard to quantitatively evaluate edge detection-based methods for range images.

Nevertheless, we were able to generate the ground truth of edge maps from the region-based ground truth segmentation. Figure 8 shows a sample image from the ABW dataset; namely abw.test.8, where we can see in Figure 8.b and Figure 8.c the ground truth of regions and the edges we have generated from it.

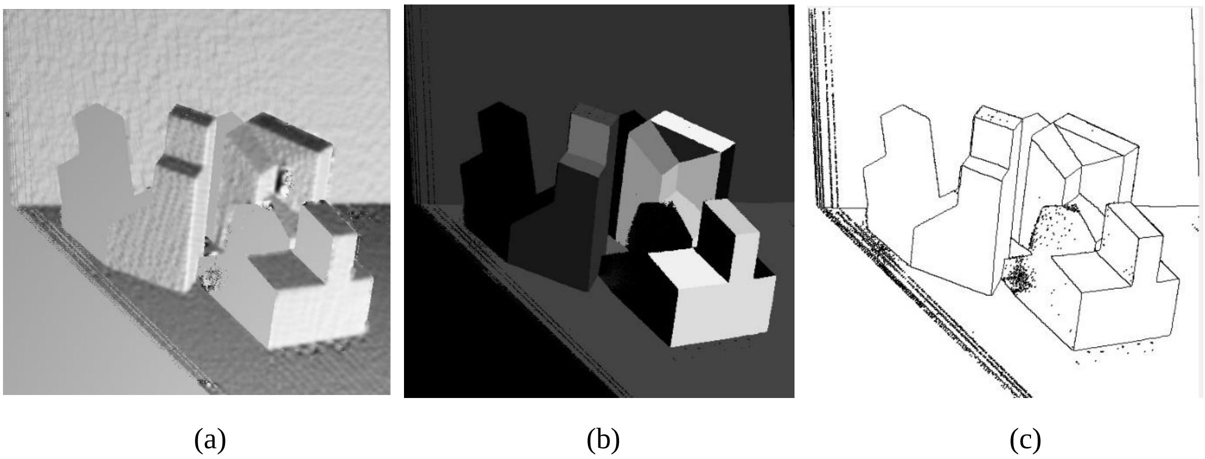


Figure 8. (a) A sample image from ABW dataset, (b) Ground truth of regions, (c) Generated ground truth of edges.

After having extracted the edge map from the ground truth region-based segmentation, provided by the ABW dataset, we consider the Dice index as a metric to evaluate the quality of the edge detection and to compare the results obtained by the proposed detector with those of the original Canny detector.

The Dice index [33] allows to express the gap between an edge map, produced by our detector and the corresponding ground truth edge map. This index is calculated based on the following elements:
True Positives (*TP*): Number of correctly detected edge pixels.

False Positives (*FP*): Number of pixels wrongly detected as edge pixels (absent in ground truth).
True Negatives (*TN*): Number of true edge pixels that were not detected.
Depending on *TP*, *FP* and *TN*, the Dice index is expressed as follows:

$$\kappa = \frac{2\times TP}{2\times TP+FP+TN} \tag{7}$$

The Dice index for the edge map produced in Figure 9.a regarding the ground truth edge map in Figure 9.b is 0.8642, where the three parameters were as follows : *TP* = 12353 pixels, *FP* = 3214 and *TN* = 666.

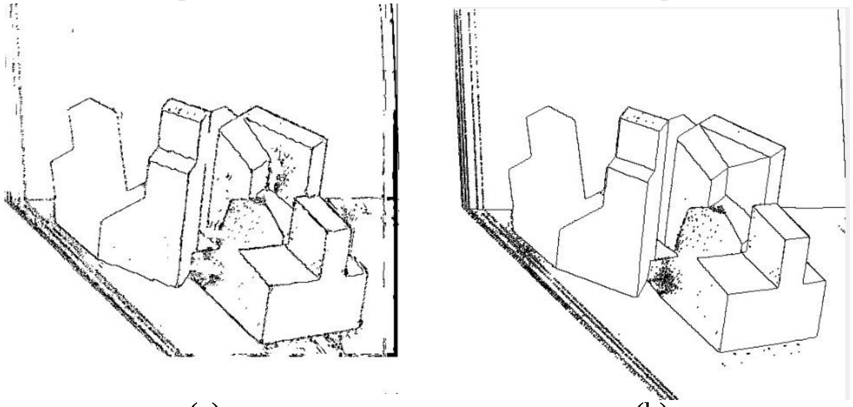


Figure 9. Comparison between detected edges and ground truth: (a) Detected edges by the proposed detector, (b) Ground truth edges.

For the whole 30 images of the ABW database, the results according to the Dice index are introduced in Table 2.

Table 2. Mean, max. and min. Dice index for the whole ABW dataset.

Mean Dice	Standard-deviation	Max. Dice	Min. Dice
0.8238	0.0335	0.8642 Obtained with abw.test.8	0.7629 Obtained with abw.test.0

Figure 10 shows the visual detection results for both images having scored highest and lowest; namely, abw.test.8 and abw.test.0.

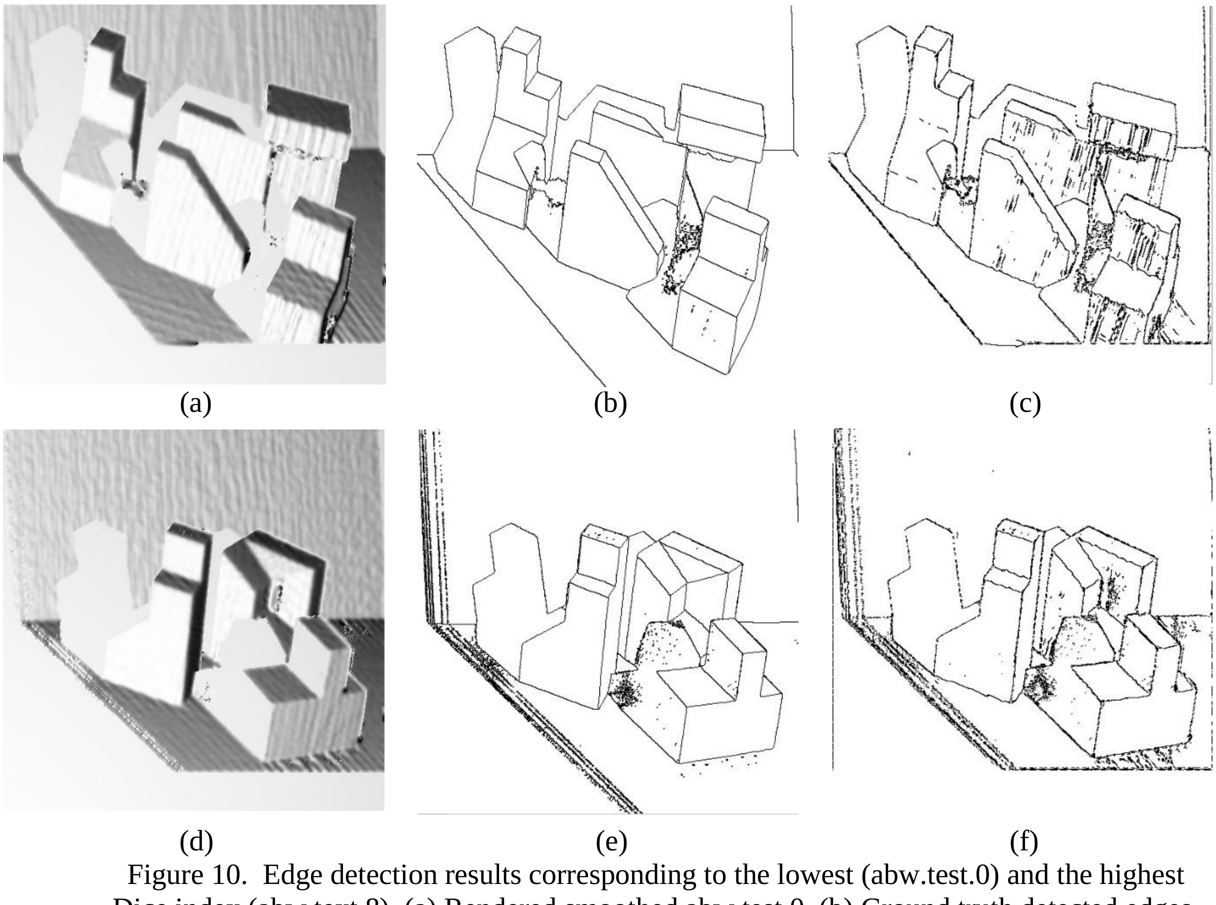


Figure 10. Edge detection results corresponding to the lowest (abw.test.0) and the highest Dice index (abw.text.8), (a) Rendered smoothed abw.test.0, (b) Ground truth detected edges, (c) Detected edges by the proposed detector, (d) Rendered smoothed abw.test.8, (e) Ground truth detected edges and (f) Detected edges by the proposed detector.

Table 3. Detection result comparison according to Dice index, involving the proposed detector and the original Canny detector.

	Mean Dice	Standard Deviation	Max. Dice	Min. Dice
Proposed detector on surface-based image	0.8238	0.0335	0.8642	0.7629
Original Canny on raw image	0.6972	0.0539	0.7813	0.5998

According to the results introduced in Table 3, we can affirm that the proposed detector for range images has considerably enhanced the detection of the edges in range images. For the mean value of the Dice index, the enhancement is about 18%, which will be very helpful for the post-processing of range images. Moreover, it is more stable, given that its standard deviation is significantly lower than that of the original Canny detector.

5. CONCLUSION

Segmenting range images, given highly noisy nature of the latter, is considered as a hard task. In practice, the nature of the processing and the nature of the image make certain segmentation methods advantageous over others. In this paper, a new edge detector for range images is proposed. The main contribution consists in an adaptation of some steps of the original Canny detector, so the new detector can be appropriately applied for edge detection in range images. Mainly, we have introduced a new method for gradient calculation after having noticed that the classical derivative calculation based on raw data cannot be applied to range images. This is due to discontinuity between different objects or between surfaces of the same object. Therefore, instead of using raw data, gradient calculation is based on a new generated surface-based image that allows to compute angles between normal vectors and object surfaces. By using such angles, we calculate new gradient norms and directions and use them for the further steps of the Canny algorithm. Like most edge detectors, our new detector is fast and can be used for real-time applications. Experimental results and their comparison with those obtained by the original Canny algorithm have allowed to state that the proposed detector is efficient and well-suited for range images. In future work, combination of range data (Depth) and color data (RGBD images) could be considered to further test and evaluate the proposed new detector.

REFERENCES

[1] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. W. Bowyer, D. W. Eggert, A. W. Fitzgibbon and R. B. Fisher, "An Experimental Comparison of Range Image Segmentation Algorithms," IEEE Trans. on Pattern Analysis and Machine Intell., vol. 18, no. 7, pp. 673–689, 1996.

[2] C. S. Won, R. M. Gray and M. Robert, Stochastic Image Processing, Springer Science & Business Media, ISBN 978-1-4419-8857-7, 2004.

[3] H. Bustince, M. Pagola, A. Jurio, E. Barrenechea, J. Fernández, P. Couto and P. Melo-Pinto, "A Survey of Applications of the Extensions of Fuzzy Sets to Image Processing," Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition, Springer, Heidelberg, vol. 256, pp. 3-32, 2009.

[4] M. Versaci and F. C. Morabito, "Image Edge Detection: A New Approach Based on Fuzzy Entropy and Fuzzy Divergence," International Journal of Fuzzy Systems, vol. 23, no. 5, pp. 918–936, 2021.

[5] J. Fang, H. Liu, L. Zhang, J. Liu and H. Liu, "Region-edge-based Active Contours Driven by Hybrid and Local Fuzzy Region-based Energy for Image Segmentation," Information Sciences, vol. 546, no. 6, pp. 397-419, 2021.

[6] T.J. Fan, G.G. Medioni and R. Nevatia, "Segmented Description of 3-D Surfaces," IEEE Journal on Robotics and Automation, vol. 3, no. 6, pp. 527–538, 1987.

[7] X. Jiang and H. Bunke, "Edge Detection in Range Images Based on Scan Line Approximation," Computer Vision and Image Understanding, vol. 73, no. 2, pp. 183–199, 1999.

[8] M. Basu, "Gaussian-based Edge-detection Methods: A Survey, " IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), vol. 32, no. 3, pp. 252–260, 2002.

[9] Y. Ding, X. Ping, M. Hu and D. Wang, "Range Image Segmentation Based on Randomized Hough Transform," Pattern Recognition Letters, vol. 26, no. 13, pp. 2033–2041, 2005.

[10] A. Bab Hadiashar and N. Gheissari, "Range Image Segmentation Using Surface Selection Criterion," IEEE Transactions on Image Processing, vol. 15, no. 7, pp. 2006–2018, 2006.

[11] D. Holz and S. Behnke, "Fast Range Image Segmentation and Smoothing Using Approximate Surface Reconstruction and Region Growing," Intelligent Autonomous Systems, vol. 12, pp. 61–73, 2013.

[12] D. Holz and S. Behnke, "Approximate Triangulation and Region Growing for Efficient Segmentation and Smoothing of Range Images," Robotics and Autonomous Systems, vol. 62, no. 9, pp. 1282–1293, 2014.

[13] S. Gupta, R. B. Girshick, P. Andres Arbelaez and J. Malik, "Learning Rich Features from RGB-D Images for Object Detection and Segmentation," Proc. of the European Conference on Computer Vision, arXiv:1407.5736, pp. 345–360, 2014.

[14] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez and J. Garcia-Rodriguez, "A Survey on Deep Learning Techniques for Image and Video Semantic Segmentation, " Applied Soft Computing, vol. 70, pp. 41–65, 2018.

[15] B. Parvin and G. Medioni, "Segmentation of Range Images into Planar Surfaces by Split and Merge," Computer Vision Pattern Recognition, pp. 415-417, 1986.

[16] R. W. Taylor, M. Savini and A. P. Reeves, "Fast Segmentation of Range Imagery into Planar Regions," Computer Vision, Graphics and Image Processing, vol. 45, no. 1, pp. 42-60, 1989.

[17] A. V. Bhavsar and A. N. Rajagopalan, "Inpainting Large Missing Regions in Range Images," Proc. of the 20th IEEE International Conference on Pattern Recognition, pp. 3464-3467, Istanbul, Turkey, 2010.

[18] P. J. Besl and R. C. Jain, "Segmentation through Variable-order Surface Fitting," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 2, pp. 167-192, 1988.

[19] N. Yokoya and M. D. Levine, "Range Image Segmentation Based on Differential Geometry: A Hybrid Approach," IEEE Trans. on Pattern Analysis and Machine Intell., vol. 11, no. 6, pp. 643-649, 1989.

[20] T. Kasvand, "The k1k2 Space in Range Image Analysis," Proc. of the 9th IEEE International Conference on Pattern Recognition, IEEE Computer Society, pp. 923-926, Rome, Italy, 1988.

[21] A. Gupta and R. K. Bajcsy, "Integrated Approach for Surface and Volumetric Segmentation of Range Images Using Biquadrics and Superquadrics," Applications of Artificial Intelligence X: Machine Vision and Robotics, International Society for Optics and Photonics, vol. 1708, pp. 210-227, 1992.

[22] A. Gupta and R. Bajcsy, "Volumetric Segmentation of Range Images of 3D Objects Using Super Quadric Models," CVGIP: Image Understanding, vol. 58, no. 3, pp. 302-326, 1993.

[23] X. Jiang and H. Bunke, "Fast Segmentation of Range Images into Planar Regions by Scan Line Grouping," Machine Vision and Applications, vol. 7, no. 2, pp. 115-122, 1994.

[24] A. Davignon, "Contribution of Edges and Regions to Range Image Segmentation," Applications of Artificial Intelligence X: Machine Vision and Robotics, International Society for Optics and Photonics, vol. 1708, pp. 228-239, 1992.

[25] Y. C. Wong, L. J. Choi, R. S. S. Singh, H. Zhang and A. R. Syafeeza, "Deep Learning-based Racing BIB Number Detection and Recognition," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 3, pp. 181-194. 2019.

[26] C. Mohamed and M. Smaine, "Edge Detection in Range Images Using a Modified Canny Filter," Proc. of the IEEE International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS), vol. 1, pp. 1-7, Skikda, Algeria, 2019.

[27] J. Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, 1986.

[28] J. P. D'Haeyer, "Gaussian Filtering of Images: A Regularization Approach," Signal Processing, vol. 18, no. 2, pp. 169-181, 1989.

[29] S. Bhardwaj and A. Mittal, "A Survey on Various Edge Detector Techniques," Procedia-Technology, vol. 4, pp. 220-226, 2012.

[30] D. J. Olive, "Multiple Linear Regression," Linear Regression Book, Springer, Cham, pp. 17-83, 2017.

[31] S. Mazouzi and Z. Guessoum, "A Fast and Fully Distributed Method for Region-based Image Segmentation," Journal of Real-time Image Processing, vol. 18, no. 3, pp. 793-806, 2021.

[32] A. Richtsfeld, T. Morwald, J. Prankl, M. Zillich and M. Vincze, "Segmentation of Unknown Objects in

Indoor Environments," Proc. of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4791–4796, Vilamoura-Algarve, Portugal, 2012.

[33] L. R. Dice, "Measures of the Amount of Ecologic Association between Species," Ecology, vol. 26, no. 3, pp. 297-302, 1945.

ملخص البحث:

تبقى تجزئة الصّور من أهم المهام، المتعلقة بتحليل الصّور وفهمها. وهي تتعامل مع الصّور الخام من أجل إعدادها لتكون قابلة للاستخدام في عمليّات أوتوماتيكية عالية المستوى؛ مثل التّصنيف أو استرجاع المعلومات.

نُقدّم في هذه الورقة كاشف حواف جديداً معدَّلاً للصّور ذات المدى. وقد استلهمنا مبدأه من كاشف (كاني)، بحيث يتمّ أخذ السِّمات المتأصّلة في الصّور ذات المدى بعين الاعتبار. وفي العادة، يُستخدم كاشف (كاني) في الصّور ذات التدرّج الرّمادي أو الصّور الملوّنة، حيث يؤدي تطبيقه المباشر مع الأعماق الى عدم الحصول على نتائج مُرضية.

من الصّورة الخام، المحتوية على الأعماق المقاسة، يتمّ حساب صّورة تخفيف تتكون من صورة للمتّجهات العمودية على السّطوح المحليّة. وهكذا يجري استخدام الزوايا بين المتجهات المتجاورة لحساب درجة ميْل بناءً على الزّوايا. وقد استخدمت صّور حقيقية من مجموعة بيانات (ABW) في عملية التجريب، إذ تفوّق الكاشف المقترح على كاشف (كاني) الأصلي بنسبة (18%).

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 03, September 2021.

WEIGHTED GREY WOLF OPTIMIZER WITH IMPROVED CONVERGENCE RATE IN TRAINING MULTI-LAYER PERCEPTRON TO SOLVE CLASSIFICATION PROBLEMS

Alok Kumar[1], Lekhraj[2] and Anoj Kumar[3]

(Received: 18-May-2021, Revised: 2-Aug.-2021, Accepted: 12-Aug.-2021)

ABSTRACT

The Grey Wolf Optimizer (GWO) is a very recently developed and emerging swarm-intelligent algorithm. The GWO algorithm was inspired by the social dominance hierarchy and hunting strategy of the grey wolves that has been successfully tailored to tackle various discrete and continuous optimization problems. During its practical implementation, however, it may be stuck in sub-optimal solutions (stagnation in local optima) due to its less exploration in the early stages that show the main drawback of this algorithm. Therefore, this research work enhances the hunting and attacking mechanism in order to modify the corresponding position updated equation and exploitation equation, respectively, to propose a novel algorithm, called Weighted Grey Wolf Optimizer with Improved Convergence Rate (WGWOIC). The effectiveness of the proposed algorithm (WGWOIC) is investigated by testing it an 33 different and fairly popular numerical benchmark functions. Although, these test functions are considered from two different benchmark datasets to assess the strength and robustness of the proposed algorithm regarding the unknown search space of the problem. In order to carry out performance analysis, moreover, the WGWOIC's results are compared against many other state-of-the-art meta-heuristic algorithms, such as Particle Swarm Optimization (PSO), Moth-Flame Optimization (MFO), Whale Optimization Algorithm (WOA), Grey Wolf Optimizer (GWO) and very recent variants of GWO. The comparative study for WGWOIC concludes that the proposed algorithm provides very competitive results against other studied meta-heuristic algorithms. Furthermore, the hybridization of the WGWOIC meta-heuristic optimization algorithm with a Multi- Layer Perceptron (MLP) neural network is employed to improve the accuracy of the classification problem. WGWOIC trainer provides the optimal values for weight and biases to the MLP network. Further, the performance is tested in terms of classification accuracy on five popular classification datasets and assesses the efficiency of the WGWOIC trainer is assessed against many other meta-heuristics trainers. The results show that the proposed algorithm eventually provides very competitive outcomes, implying that the WGWOIC algorithm offers a better exploitation, explores the search space and effectively solves several different classification problems.

KEYWORDS

Meta-heuristic algorithms, Evolutionary algorithms, Nature-inspired algorithms, Swarm-based algorithms, Grey wolf optimizer, Neural network, Multi-layer perceptron.

1. INTRODUCTION

Real-world problems have an unknown search space with their unknown solution. Besides, only limited resources and limited time are available to tackle these problems. Therefore, an optimum solution should exist to resolve the above issues and overcome the limitations. Consequently, there should be the existence of such algorithms provided likely to the optimum solution. The optimization algorithms are able to fulfill the above requirements and overcome the above limitations. The optimization algorithms' particular category is called meta-heuristic algorithms, becoming more popular in the last two decades due to their simplicity and flexibility. Its processing begins with random solution(s) and ends at the optimum solution(s), making it more robust. In other words, the meta-heuristic algorithms' initial solutions are known as random solutions; evolve these random solutions evolve through the applied well-known algorithms and obtain final solutions known as optimum solutions. To sum up, the researchers' main aim is to design various new meta-heuristic optimization algorithms and enhance the existing algorithms to obtain global optimum solutions. Surprisingly, the No Free Lunch (NFL) theorem [1] states that a specific optimization algorithm cannot extensively tackle all types of problems; however, not all

A. Kumar, Lekhraj and A. Kumar are with the Department of Computer Science and Engineering, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, UP, India, Emails: [1]alokkumar@mnnit.ac.in,[2]lekhraj@mnnit.ac.in and [3]anojk@mnnit.ac.in

kinds of optimization algorithms can solve a single problem. Consequently, this encourages designing various new meta-heuristic optimization algorithms and improving existing algorithms.

Five types of optimization algorithms come under meta-heuristic algorithms: evolutionary algorithms, bio-stimulated algorithms, physics-based algorithms, nature-inspired algorithms and swarm intelligence-based algorithms. The hierarchical diagram of various meta-heuristic algorithms is depicted in Figure 1. Holland proposed the Genetic Algorithm (GA) [2] in 1992 and it is the most famous algorithm of evolutionary class. The GA algorithm is typically justified by the Darwin's evolution theory. The first real-world application of GA was c ontrol system optimization using genetic algorithms, which was proposed by Krishnakumar and Goldberg [3]. Subsequently, the various other evolutionary algorithms such as Differential Evolution (DE) [4] algorithm, Biogeography-Based Optimizer (BBO) [5], Evolutionary Programming (EP) [6], Genetic Programming (GP) [7], …etc., came into the picture. In addition to the above evolutionary algorithms, Covariance Matrix Adaptation (CMA-ES) [8] and Fast Evolutionary Programming (FEP) [9] are other evolutionary meta-heuristic algorithms. In the second class, the Artificial Immune System [10], Bacterial Foraging Optimization (BFO) [11], …etc., come under bio-stimulated algorithms.

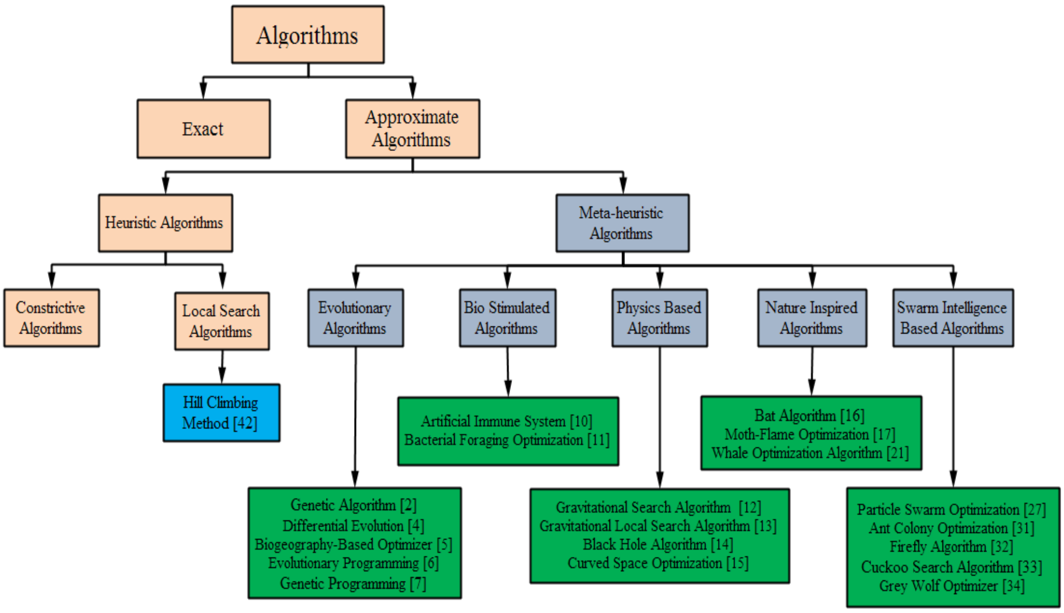

Algorithms
Exact
Approximate Algorithms
Heuristic Algorithms
Meta-heuristic Algorithms
Constrictive Algorithms
Local Search Algorithms
Hill Climbing Method [42]
Evolutionary Algorithms
Bio Stimulated Algorithms
Physics Based Algorithms
Nature Inspired Algorithms
Swarm Intelligence Based Algorithms
Artificial Immune System [10]
Bacterial Foraging Optimization [11]
Bat Algorithm [16]
Moth-Flame Optimization [17]
Whale Optimization Algorithm [21]
Genetic Algorithm [2]
Differential Evolution [4]
Biogeography-Based Optimizer [5]
Evolutionary Programming [6]
Genetic Programming [7]
Gravitational Search Algorithm [12]
Gravitational Local Search Algorithm [13]
Black Hole Algorithm [14]
Curved Space Optimization [15]
Particle Swarm Optimization [27]
Ant Colony Optimization [31]
Firefly Algorithm [32]
Cuckoo Search Algorithm [33]
Grey Wolf Optimizer [34]

Figure 1. Hierarchical representation of meta-heuristic optimization algorithms.

The third class of meta-heuristic algorithms is referred to as the physics-based algorithms that mimic the physics rules. The Gravitational Search Algorithm (GSA) [12], Gravitational Local Search Algorithm (GLSA) [13], Black Hole (BH) [14] algorithm, Curved Space Optimization (CSO) [15], …etc., come under these algorithms. Subsequently, the Bat Algorithm (BA) [16], Moth-Flame Optimization (MFO) [17], Whale Optimization Algorithm (WOA) [21], …etc., are listed with nature-inspired algorithms that are the fourth category of meta-heuristic algorithms. In addition, the above algorithms are used to tackle real-world applications; for instance, Abed-alguni [22] introduced a novel Q-learning approach using the bat algorithm that finds optimal Q-values and validated the performance on the shortest path problem and the taxi problem. Besides the algorithms mentioned earlier, the State of Mater Search (SMS) [23], Flower Pollination Algorithm (FPA) [24], …etc., are called population-based algorithms that are inspired by a different source.

The last but not most minor class is swarm intelligence-based algorithms, inspired by the intelligent swarm behavior. A large group of homogeneous living species is called a swarm, such as bird flocking and fish schooling. The Particle Swarm Optimization (PSO) [27] is the most famous swarm-based algorithm that mimics the social behavior of birds. Kennedy and Eberhart proposed this algorithm in 1995. In addition to the PSO, Ant Colony Optimization (ACO) [31], Firefly Algorithm (FA) [32], Cuckoo Search Algorithm (CSA) [33], Grey Wolf Optimizer (GWO) [34], …etc., are likely grouped into swarm-based algorithms.

The GWO is a novel and emerging swarm-intelligent algorithm inspired by the grey wolves' social dominance hierarchy and hunting strategy. The GWO has been successfully tailored to tackle various discrete and continuous optimization problems. The main drawback of the GWO is that it may be stuck in sub-optimal solutions (stagnation in local optima) due to its less exploration in the early stages. Nevertheless, the exploration and exploitation should be adequately balanced to extensively investigate the search space for achieving the most optimal solutions. In order to overcome the above limitations, Mittal et al. [35] proposed an advanced variant of GWO; namely, Modified Grey Wolf Optimizer (mGWO). This variant offers a pertinent equilibrium among exploration and exploitation for the search space; however, the modification is attempted in only controlling parameter $\vec{a}$; hence, further improvement may be possible. Furthermore, Singh [36] enhanced the biological order of the social hierarchy of grey wolves in order to propose another variant of GWO. However, this research work improved only the biological structure regarding the social hierarchy of grey wolves, but not the significant enhancement in the mathematical model accordingly. In addition, Kumar et al. [38] proposed another variant of GWO, named WMGWO. This research work allocated the static weight instead of dynamic weights to the alpha, beta and delta search agents. Hence, the above limitations and drawbacks motivate the researcher, to propose another novel variant of GWO.

In brief, our main contributions are as follows:

- The position update equation has been modified to enhance the hunting behavior of grey wolves in order to propose a novel algorithm of GWO to overcome the above limitations and shortcomings of the basic GWO and its very recant algorithms.
- In addition, the exploitation equation is adopted [35] to enhance the encircling and attacking mechanisms.
- This research work considers 33 mathematical benchmark functions from two different datasets to examine the effectiveness and justify the robustness of the proposed algorithm.
- The results are compared against various state-of-the-art meta-heuristics optimization algorithms: MFO [17], WOA [21], PSO [27], GWO [34], mGWO [35], MVGWO [36] and WMGWO [38].
- Furthermore, the proposed algorithm has been applied to optimize the weights and biases to train the MLP solving real-world classification problems considering five datasets and the results were compared with those of several well-known meta-heuristic trainers that eventually offer salutary influence against unknown search space.

1.1 Roadmap

The remaining sections of this paper are organized in the following way. The MFO, WOA, PSO and GWO and their many recent variants are discussed in Section 2 that addresses the limitations and shortcomings associated with grey wolf optimizers and their very recant variants, which eventually motivates to propose another enhanced algorithm of GWO. Section 3 describes the key functionality of basic GWO and a comprehensive discussion about the suggested novel algorithm. Section 4 discusses the performance results and experimental analysis of the proposed algorithm against state-of-the-art meta-heuristic algorithms. The proposed algorithm employed with a multi-layer perceptron is presented in Section 5. Finally, Section 6 concludes this research work and provides a future research direction.

2. Literature Review

This section elaborates a comprehensive discussion regarding various state-of-the-art meta-heuristic optimization algorithms. The MFO, WOA, PSO, GWO and very recent algorithms of GWO will be the key focus in this study; however, the applied potential applications also will be extensively discussed.

The Moth-Flame Optimization (MFO) [17] is the nature-inspired optimization algorithm that was proposed by Mirjalili in 2015. The transverse orientation is the inspiration to introduce this algorithm. The computational cost of MFO is O(t*n2 + t*n*d), in which 't' refers to the maximum number of iterations, 'n' indicates the number of moths and 'd' is the number of variables. This algorithm is tested on 29 benchmark functions and seven real-engineering problems. The effectiveness of the proposed algorithm is validated using the comparison of results against PSO, GSA, BA, FPA, SMS, FA and GA. Consequently, this algorithm was able to outperform the comparative algorithms against the majority of test functions. The other very recent variants of MFO are proposed in [18]-[20].

The Whale Optimization Algorithm (WOA) [21] is the most recent meta-heuristic optimization algorithm. Mirjalili and Lewis proposed this algorithm in 2016. This proposed algorithm is another nature-inspired algorithm and is motivated by the bubble-net hunting strategy of humpback whales. The whale is a mammal, implying that whales provide milk for their children and are recognized as giant animals on earth. To validate the performance of this algorithm, it is benchmarked on 29 optimization test functions and six engineering-design problems. The welded beam design, tension/compression spring design, pressure vessel design and other three bar truss design problems are considered to be solved using this algorithm. In bar truss design, 52-bar truss design, 25-bar truss design and 15-bar truss design are taken into account. The comparative analysis of results has validated the effectiveness of WOA against PSO, GSA, DE and CMA-ES / FEP. The Binary WOA [25] and another very recently enhanced algorithm of WOA with the map-reduced application are proposed in [26].

The Particle Swarm Optimization (PSO) [27] is a Swarm Intelligence (SI) meta-heuristic optimization algorithm. It was the first social behavior-based algorithm proposed by Kennedy and Eberhart in 1995. The algorithm mimics the social intelligence of bird flocking and fish schooling. The practical execution of the PSO algorithm starts with random solutions (called initial population), which are optimized over the course of iterations using PBEST (Personal best) and GBEST (Global best) parameters. The velocity and position vectors are the mathematical parameters, whereas inertia weight, the cognitive component and social components are the tuning parameters of this algorithm. The several other latest improved algorithms of PSO named UPSO [28] and population size in PSO are referenced in [29]. In addition, Alshdaifat and Bataineh [30] enhanced the PSO, named improved PSO and further employed it with Chebyshev distribution (which defines the search space for IPSO) for optimizing and thinning of the planar array.

Mirjalili et al. [34] developed, theoretically defined and programmatically implemented the Grey Wolf Optimizer (GWO) in 2014. The GWO is a genuinely emerging meta-heuristic optimization algorithm in the literature. The grey wolves are found in Eurasia and North America called *Canis lupus*. The grey wolves' social structure and hunting mechanism were cited as influences for this approach. Figure 2 depicts the dominant social hierarchy. According to Figure 2, the pack's all grey wolves are categorized into four categories corresponding to their specific dominancy and pursuit role. The figure illustrates that the alpha wolf is at the top of the dominant social hierarchy. This wolf is the pack's leader and is referred to as the manager of the pack. The beta wolf is the pack discipliner and the alpha's counselor who endures the next step down. On the social hierarchy's third level, the delta wolf is located that is sentinel, hunter, advisor to beta and caretaker to the pack. As the group's helpers and babysitters, the omega wolves are left. The specific types' hunting method incorporates three pivotal steps that refer to additional motivation rather than the social hierarchy. As a result, the primary phases are to seek the prey and annoy the prey until it gives up or stops and then attack the target in the end. The mathematical model regarding the above lemma has been formulated to introduce the GWO algorithm. In addition, on 29 test functions and four engineering-design real-world's problems, performance has been tested and certified. In addition, the PSO, GSA, DE and FEP algorithms were compared to the GWO in order to verify the findings of this work. Consequently, the experimental results determine the effectiveness of this algorithm that produces very competitive results. However, this algorithm may be stuck in local optima that refer to its main drawback. In addition, the poor solution accuracy and sluggish convergence rate address it more challenging for further improvement.

Mittal et al. [35] proposed an advanced variant of GWO; namely, modified Grey Wolf Optimizer (mGWO), in order to maintain pertinent equilibrium among exploration and exploitation of the search space. In order to accomplish the focus on objective, they employed the exponential decay function instead of the linear function pertaining to the constant vector $\vec{a}$ in the enforcement of the standard algorithm of GWO. The exponential function devotes seventy and thirty percent iterations to exploration and exploitation, respectively. In comparison, in order to accomplish the linear function, half of the iteration; i.e., the first fifty percent is dedicated to the exploration and the remaining fifty percent is committed to exploitation. Note that multimodal, unimodal, composite and fixed-dimension multimodal benchmark functions are imposed to illuminate the proposed variant's performance, considering that standard deviation and average are statistical parameters of appraisal with 3000 iterations and 30 number of population. In addition, the selection of cluster heads in wireless sensor networks is often regarded as a relatively well-known real-world application. To sum up, the mGWO outperforms occasionally or has very competitive outcomes compared to other meta-heuristic algorithms and original GWO. However, the modification is attempted in only controlling parameter $\vec{a}$, hence further improvement may be possible.

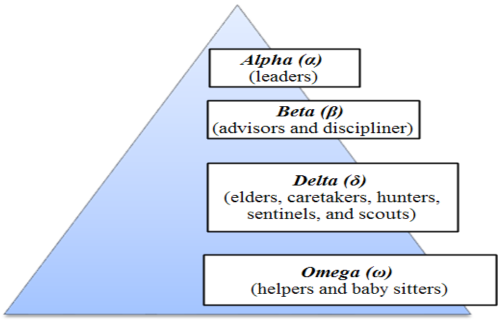

Alpha (α)
(leaders)
Beta (β)
(advisors and discipliner)
Delta (δ)
(elders, caretakers, hunters, sentinels, and scouts)
Omega (ω)
(helpers and baby sitters)

Figure 2. Social dominant hierarchy of grey wolves (dominance decreases from top down) [34].

The MVGWO is another variant of GWO proposed by Singh [36] in 2018. There was a biological improvement in the social hierarchy, in which five groups are formed for the total population of wolves. Thus, the proposed algorithm extends the social order up to five levels after including gamma wolves at the third level from the top. According to biological theory, the top four levels' wolves (i.e., delta, gamma, beta and alpha) participated in the hunting and finding of prey. In order to carry out mathematical implementation, encircling behavior and position update equation has been modified in terms of basic GWO to improve the results. The average of the best four solutions is utilized to update the remaining solutions over the course of iteration and find the most optimal solution at the end of the last iteration. The obtained results show that this algorithm provides very competitive results concerning PSO, GWO and modified mean GWO [37]. In addition, the newly modified algorithm performs considerably better to tackle the cantilever beam design problem and sine dataset. This research work improved only the biological structure of the grey wolves, but it has not led to significant enhancement in the mathematical model accordingly.

Kumar et al. [38] proposed another variant of GWO, named WMGWO, in 2019. There are four levels of social hierarchy. The proposed variant employed a weighted mean factor instead of uniform distribution in order to update the omegas. It suggested 54, 30 and 16 percent weightage to alpha, beta and delta wolves (i.e., search agents or solution), respectively. The performance of the proposed variant is validated after comparing the results with these of GWO, mGWO and MVGWO. The outcomes of this algorithm are very competitive against comparative algorithms. In addition, this algorithm performs very well on the function approximation and classification datasets. This research work utilized static weight instead of dynamic weights to the alpha, beta and delta search agents, which pointed to the limitation of this work. To improve the diversity of GWO, Abed-alguni and Barhoush [39] introduced a distributed approach of GWO by organizing its population using the island model. Furthermore, the proposed algorithm was tested on thirty CEC 2014 functions and fifteen standard test functions that provide competitive performance against other tested algorithms.

3. Proposed Work

The basic Grey Wolf Optimizer (GWO) and the novel proposed algorithm (Weighted Grey Wolf Optimizer with Improved Convergence Rate (WGWOIC)) will be discussed in this section.

3.1 Grey Wolf Optimizer

As we discussed in the preceding section, the grey wolf optimizer [34] mimics the social dominant hierarchy of grey wolves and their social hunting mechanism inspires this algorithm. According to the biological theory of grey wolves, hunting is attempted exclusively *via* the top three levels' wolves (i.e., alpha, beta and delta). The alpha wolf is the dominant wolf, then beta and then delta. All other wolves dominate the omega wolves in the pack. The social dominant hierarchy of grey wolves has been depicted in Figure 2. In order to solve any optimization problem using GWO, the process commences with the random population (also designated as random search agents or random solutions). Subsequently, this algorithm's workflow would originate. For the mathematical model of GWO, the best three solutions obtained so far are saved and remaining solutions (including the omegas) are adapted based on the above three best search agents according to Equation 1.

$$\vec{X}(\text{Curr\_iter}+1)=\frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3} \tag{1}$$

According to Equation 1, Curr_iter refers to the current iteration value that linearly increases to Max_iter (maximum number of iterations). The vectors $\overrightarrow{X_1},\overrightarrow{X_2}\ and\ \overrightarrow{X_3}$ indicate the updated best positions (solutions) of alpha, beta and delta wolves, respectively. These three wolves update their positions according to the prey's position that is formulated by Equation 2. However, there is no idea regarding the location of the prey (optimum) in an abstract search space. Therefore, alpha, beta and delta determine the prey's probable location by taking the mean of their positions. The mathematical result of the mean is represented by $\vec{X}(Cur\_iter+1)$, on which basis the remaining wolves (omegas) update their positions. In order to determine the preceding best three positions, $\overrightarrow{D_\alpha},\overrightarrow{D_\beta}\ and\ \overrightarrow{D_\delta}$ vectors have to be figured out using Equation 2, whereas the vector $\vec{\mathrm{D}}$ indicates the distance from wolf to prey.

$$\left.\begin{array}{l}\overrightarrow{D_\alpha}=\left|\overrightarrow{C_1}.\overrightarrow{X_\alpha}-\vec{X}\right|,\overrightarrow{X_1}=\overrightarrow{X_\alpha}-\overrightarrow{A_1}.(\overrightarrow{D_\alpha})\\ \overrightarrow{D_\beta}=\left|\overrightarrow{C_2}.\overrightarrow{X_\beta}-\vec{X}\right|,\overrightarrow{X_2}=\overrightarrow{X_\beta}-\overrightarrow{A_2}.(\overrightarrow{D_\beta})\\ \overrightarrow{D_\delta}=\left|\overrightarrow{C_3}.\overrightarrow{X_\delta}-\vec{X}\right|,\overrightarrow{X_3}=\overrightarrow{X_\delta}-\overrightarrow{A_3}.(\overrightarrow{D_\delta})\end{array}\right\} \tag{2}$$

The vectors $\vec{A}$ and $\vec{C}$ are called controlling parameters that provide equilibrium among exploration and exploitation for the abstract search space. The value of these controlling parameters is calculated by using Equation 3.

$$\vec{A}=2*\vec{a}.\overrightarrow{r_1}-\vec{a},\vec{C}=2.\overrightarrow{r_2} \tag{3}$$

The vectors $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ are known as random vectors between [0, 1]; i.e., the computational value of $\vec{C}$ would be found between [0, 2] and the value of $\vec{A}$ between [-2, 2]. The vector $\vec{C}$ is deliberately used to provide a random value throughout the algorithm, which offers randomness to the GWO algorithm. In contrast, if vector $\vec{A}$'s mathematical weight is found in the interval [-1, 1], it supports the algorithm to converge toward the most optimal solution; otherwise, it diverges from the current solution in order to find the optimum. In addition to the above vectors, another vector,$\vec{a}$, is also called a controlling parameter which is calculated by Equation 4. The value of this controlling parameter is decreased linearly to 0 over the course of iterations. Therefore, the initial fifty percent values oblige exploration and the remaining fifty percent is devoted to exploitation.

$$\vec{a}=2*\left(1-\frac{\text{Curr\_iter}}{\text{Max\_iter}}\right) \tag{4}$$

The GWO algorithm was tested on twenty-nine benchmark functions and the obtained results were analyzed to check the performance. In order to investigate the performance, the comparative analysis against other well-known algorithms asserts that the GWO algorithm encounters some limitations and challenges, such as stagnation in local optima, low solving accuracy and slow convergence rate. Hence, these limitations and challenges encourage proposing another algorithm of GWO. Therefore, we have introduced another algorithm of GWO titled Weighted Grey Wolf Optimizer with Improved Convergence Rate (WGWOIC), as discussed in the coming sub-section.

3.2 Proposed WGWOIC Algorithm

As we have discussed earlier, the basic GWO and its very recently developed algorithms have some limitations and shortcomings. Therefore, we have proposed another algorithm of GWO to overcome these limitations and resolve the deficiencies. The proposed algorithm is designated as Weighted Grey Wolf Optimizer with Improved Convergence Rate (WGWOIC). Therefore, we have modified the hunting (position update equation) and attacking (exploitation equation) behavior of grey wolves to introduce this algorithm that indicates this research work's novelty. In addition to the above contribution, we have practiced two different benchmark datasets to determine the effectiveness of performance in terms of strength and robustness of the proposed algorithm against other comparative state-of-the-art optimization algorithms.

According to the advancement in the biological theory of grey wolves to introduce this novel algorithm, the wolves hunt the prey and attack the target with a different mechanism. For the mathematical model, the position update equation (hunting) and exploitation equation (attacking the prey) are modified to assist the enhancement of the studied algorithm in the literature. Therefore, Equation 1 obliges hunting the prey; i.e., it is known as the hunting equation or position update equation. In contrast, Equation 4 obliges attacking the target; i.e., an attacking equation or exploitation equation. In order to accomplish our objective, we will apply Equation 5 instead of Equation 1 and Equation 7 instead of Equation 4.

$$\vec{\mathrm{X}}(\mathrm{Curr\_iter}\ +1) = (\overrightarrow{\mathrm{w_1}} + \vec{\mathrm{r}}).\overrightarrow{\mathrm{X_1}} + (\overrightarrow{\mathrm{w_2}}.\overrightarrow{\mathrm{X_2}} - \overrightarrow{\mathrm{w_3}}.\overrightarrow{\mathrm{X_3}}) \tag{5}$$

The vector $\vec{r}$ is a random weight granted to the alpha solution computed as half of the random value over every course of iteration. This random weight obliges the computed solution to be slightly tilted toward the alpha, because the alpha is likely to be considered the best solution to the problem. It is brightly founded from the comprehensive literature that alpha, beta and delta wolves contain an excellent knowledge of prey (solution) against remaining wolves. Simultaneously, the alpha wolf comprises the most optimal solution among the population, then beta and then delta. We have utilized this lemma and accordingly updated the hunting and attacking mechanisms of the wolves. Therefore, the alpha wolf is not delivering decisions alone for hunting, staying and other pack activities. Thus, the alpha wolf has also taken the advice of beta and delta. For the mathematical model, these are the three best solutions among all solutions obtained so far. In this research work, we are giving extra weight to the alpha solution to update the solutions; alongside, we have also added a weighted difference of beta and delta solutions. Consequently, the updated solutions are slightly tilted toward the alpha solution and the weighted difference provides diversity to the proposed algorithm. In Equation 5, the vectors $\overrightarrow{w_1}$, $\overrightarrow{w_2}$ and $\overrightarrow{w_3}$ are considered as influence factors that provide the influence weighted to alpha, beta and delta solutions, respectively, during every course of iteration. These factors are mathematically formulated by Equation 6.

$$\overrightarrow{w_1} = \frac{\overrightarrow{r_1}}{\overrightarrow{r_1}+\overrightarrow{r_2}+\overrightarrow{r_3}}, \overrightarrow{w_2} = \frac{\overrightarrow{r_2}}{\overrightarrow{r_1}+\overrightarrow{r_2}+\overrightarrow{r_3}}, \overrightarrow{w_3} = \frac{\overrightarrow{r_3}}{\overrightarrow{r_1}+\overrightarrow{r_2}+\overrightarrow{r_3}} \tag{6}$$

It is clear that the total sum value of these influence factors is '1' $(\overrightarrow{w_1} + \overrightarrow{w_2} + \overrightarrow{w_3} = 1)$. In order to calculate the above influence vectors, the vectors $\overrightarrow{r_1}$, $\overrightarrow{r_2}$ and $\overrightarrow{r_3}$ are utilized with the random weights in [0, 1] to provide the randomness to the proposed algorithm; not only the initial iteration, even till the final iteration. The influence factors' values will assist to find the most optimal location of the prey after putting these values in Equation 5.

In addition to the above contribution, we have adopted Equation 7 from reference [35] in order to compute the value of vector $\vec{a}$. Now, we will use Equation 7 instead of Equation 4 to enhance the exploitation of the recommended algorithm.

$$\vec{a} = 2 * \left(1 - \frac{\mathrm{Curr\_iter}^2}{\mathrm{Max\_iter}^2}\right) \tag{7}$$

From the comprehensive literature, the grey wolves accomplish their hunt by attacking the prey when it stops moving. In a mathematical model, the vector $\vec{a}$ performs this task. The value of this vector is decreased exponentially from '2' to '0' over the course of iterations. The initial seventy percent values of this vector that decrease slowly oblige extensive exploration of the search space, whereas the remaining thirty percent components that decrease quickly oblige fast exploitation toward the solution. This vector eventually maintains good equilibrium among exploration and exploitation of the abstract search space; i.e., it provides fast convergence and more diversity. This vector is also utilized to decrease randomness; implying that the WGWOIC algorithm would be converging toward the final solution. The pseudo-code of the proposed algorithm is depicted in Figure 3.

4. Results and Discussion

In this section, we have benchmarked the WGWOIC algorithm on 33 fairly well-known numerical benchmark test functions. The first twenty-three classical functions are included from

```
Initialize the population of grey wolves (Search Agents) $\overrightarrow{X_k}$ (k = 1, 2, 3... n)
Initialize controlling parameters $\vec{a}$, $\vec{A}$, and $\vec{C}$
Compute the fitness of each search agent
$\overrightarrow{X_\alpha}$ = the most fittest solution from all search agents
$\overrightarrow{X_\beta}$ = the second fittest solution from all search agents
$\overrightarrow{X_\delta}$ = the third fittest solution from all search agents
while (Curr_iter <= Max_iter) do
    for each search agent ($\overrightarrow{X_k}$) do
        Update the position of current $\overrightarrow{X_k}$ using equation (5)
    end for
    Update the value of controlling parameters $\vec{a}$ (using equation 7), $\vec{A}$, and $\vec{C}$
    Compute the fitness of all search agents
    Update the value of $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$, and $\overrightarrow{X_\delta}$
    Curr_iter = Curr_iter + 1
end while
return $\overrightarrow{X_\alpha}$
```

Figure 3. Pseudo-code of the WGWOIC algorithm.

CEC 2005, which many researchers utilized in their work. These benchmarked functions are minimization functions and categorized into three groups: unimodal (first seven functions), multimodal (following six functions) and fixed-dimension multimodal (last ten functions) benchmark functions. However, these benchmark test functions have different dimensions and boundary ranges that indicate the main challenges for the proposed algorithm in order to optimize the above test functions.

Subsequently, the remaining ten test functions are modern single objective minimization functions (CEC01 to CEC10) included from CEC-C06 2019. These test functions are scalable and known as the 100-Digit Challenge. The functions from CEC04 to CEC10 are rotated and shifted, whereas the functions from CEC01 to CEC03 are not. The dimensionality of CEC 01, CEC 02 and CEC 03 test functions is 9, 16 and 18, whereas the boundary range is [-8192, 8192], [-16384, 16384] and [-4, 4], respectively. In contrast, the dimensionality of the remaining functions (from CEC04 to CEC10) is the same, each with 10-dimensional in [−100, 100] boundary range. The detailed discussion about the first twenty-three test functions is in reference [34] and that of the remaining ten functions of CEC-C06 2019 are in [40]. At the same time, the programming implementation of CEC-C06 2019 functions is performed in reference [41]. Interestingly, this research work considers the above two different benchmark functions in order to evaluate the performance with the effect of the proposed algorithm that validates the strength and confirms the robustness of the WGWOIC algorithm.

For the mathematical implementations, the population size of the proposed algorithm and other state-of-the-art comparative algorithms is 30. All algorithms are iteratively repeated 500 times over the course of iterations to obtain the most optimal solution in one independent run. Subsequently, all algorithms are repeated 30 separate runs on each benchmark function. The average (mean) value of these 30 independent runs eventually indicates the outcome (optimum global value) to the corresponding benchmark function. The other statistical variables (Best, Worst and Std.) are also utilized to validate the effectiveness of the proposed algorithm's outcomes against the studied comparative well-known algorithms. The best statistical variable shows the minimum value throughout the 30 independent runs, whereas the worst refers to the maximum value. On the other hand, Std. stands for standard deviation, estimated through 30 separate runs. Therefore, the lowest values represent the optimum values of each statistical variable regarding all algorithms concerning individual functions. The performance of the proposed algorithm is validated against many swarm intelligence-based algorithms, such as Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Modified GWO (mGWO), Modified Variant of GWO (MVGWO) and Weighted Mean GWO (WMGWO). In addition, the proposed algorithm is also compared with two nature-inspired algorithms: Moth-Flame Optimization (MFO) and Whale Optimization Algorithm (WOA). However, the GWO has already been reached with PSO as the swarm intelligence-based algorithm, GSA as the physics-based algorithm and DE, FEP and CMA-ES as the evolutionary algorithms. Table 1 lists the simulation hardware and software environment on which the practical implementation of this work has been conducted.

Table 1. Experimental environment.

Parameter	Hardware and Software Configuration
Implementation Tool	MATLAB R2017a
Processor	Intel(R) Core(TM) i7-4770 CPU@ 3.40GHz
RAM	12.0 GB
Operating System	64-bit Operating System

Table 2 shows the computational results of unimodal test functions of the WGWOIC algorithm and other comparative state-of-the-art meta-heuristic algorithms. It must be noted that there are seven unimodal functions (F1-F7) which are having single optima. These benchmark functions validate the effectiveness of the exploitation performance of the proposed algorithm. This algorithm highly outperforms on F1, F3, F4 and F7 functions against well-known meta-heuristic comparative algorithms and provides very competitive results on the remaining unimodal functions.

Table 2. Results of unimodal benchmark functions.

Functions	Criteria	PSO	MFO	WOA	GWO	mGWO	MVGWO	WMGWO	WGWOIC
F1	Best	1.63886E-05	0.158221405	2.79122E-88	1.70234E-29	4.48066E-39	9.38977E-22	2.12829E-39	1.78217E-85
	Worst	0.000772625	20002.75312	3.06783E-69	3.2754E-27	3.39035E-34	5.58183E-20	2.9993E-36	5.13144E-81
	Mean	0.00015484	2670.895162	1.10514E-70	8.15964E-28	1.4291E-35	7.52036E-21	1.83367E-37	3.54905E-82
	Std	0.000155845	6395.635248	5.59769E-70	8.6975E-28	6.17581E-35	1.09375E-20	5.53603E-37	9.94418E-82
F2	Best	0.005416498	0.15259518	1.87787E-58	4.31376E-17	5.76217E-23	1.08145E-13	1.64269E-23	1.41081E-46
	Worst	0.122725197	60.04019074	2.18917E-50	4.50204E-16	7.94649E-21	2.57471E-12	1.02317E-21	5.6575E-44
	Mean	0.034715846	35.09009906	1.03354E-51	1.18525E-16	1.02612E-21	9.93847E-13	1.19183E-22	9.82965E-45
	Std	0.028536391	18.67345722	4.0533E-51	8.06393E-17	1.58156E-21	5.24122E-13	1.90205E-22	1.16457E-44
F3	Best	32.15428241	2669.898671	17647.46009	4.91195E-09	2.03961E-10	1.72282E-06	6.41861E-13	1.91335E-61
	Worst	184.6971148	43943.76011	72344.95817	0.001286365	5.57145E-06	0.008222075	2.37055E-06	2.20352E-55
	Mean	95.15931899	18590.53167	45553.67157	4.96083E-05	2.44554E-07	0.000462987	1.46535E-07	1.04907E-56
	Std	33.71188464	10843.57994	12483.87229	0.000234343	1.0163E-06	0.001481328	4.72723E-07	4.21886E-56
F4	Best	0.641915093	54.8705224	4.669865609	5.87068E-08	8.74413E-11	4.03672E-06	6.91339E-11	2.03365E-35
	Worst	1.787857145	81.0563014	89.38407148	2.28106E-06	7.62111E-09	0.000176699	6.02427E-09	1.21729E-31
	Mean	1.156085088	69.21414183	45.72716132	6.322E-07	1.66461E-09	3.74597E-05	9.19472E-10	8.08294E-33
	Std	0.319797238	6.824054473	27.6934708	5.65885E-07	1.91586E-09	3.84676E-05	1.25221E-09	2.54366E-32
F5	Best	27.20201917	309.3363674	27.24769259	25.74622807	26.05555255	25.9950154	26.04231519	27.24591773
	Worst	265.766625	80033040.29	28.76765556	28.55897177	28.72223185	28.80989169	28.73810387	28.90556925
	Mean	76.73511711	2686931.035	28.07688863	27.05204892	26.94454281	27.31357229	26.94364717	28.02425226
	Std	51.71636381	14608392.47	0.445358351	0.753119474	0.656530374	0.883426624	0.670585955	0.512886109
F6	Best	7.68491E-06	0.542083492	0.112093851	0.23186088	0.243501503	0.45882678	0.248906786	3.707888307
	Worst	0.00657646	10106.08889	0.959981164	1.755332071	1.255444247	3.264675357	1.507850263	4.764877793
	Mean	0.000365663	1013.161376	0.407078881	0.778026799	0.592550125	1.342234409	0.832794353	4.3288351
	Std	0.001186222	3081.722471	0.240597427	0.398519267	0.256924289	0.572752401	0.314861247	0.332820474
F7	Best	0.064203293	0.085215057	0.000116117	0.00043649	0.000379606	0.001117032	0.000518688	2.96069E-05
	Worst	0.302894078	29.62285404	0.012456871	0.006311453	0.003197678	0.005411433	0.003334457	0.001077777
	Mean	0.19398366	3.463092413	0.002654718	0.002235133	0.001484414	0.002787913	0.001564218	0.000380318
	Std.	0.064673982	6.895509972	0.00282493	0.001183375	0.000822758	0.001058371	0.000753789	0.000301542

Table 3. Results of multimodal benchmark functions.

Functions	Criteria	PSO	MFO	WOA	GWO	mGWO	MVGWO	WMGWO	WGWOIC
F8	Best	-6740.805369	-9602.385548	-12569.45849	-7442.734249	-7291.978467	-7600.083886	-7344.16222	-4427.955563
	Worst	-2871.327832	-6870.808247	-7093.47263	-3480.901369	-2868.810357	-4831.305124	-3473.562341	-3093.946616
	Mean	-4541.466559	-8394.321875	-10354.0438	-5966.223462	-5595.560325	-5733.209196	-6029.573094	-3631.225745
	Std	1141.174277	676.4760095	1819.704826	835.1905512	1210.17144	721.447048	954.272808	343.1655987
F9	Best	36.32221174	100.7700387	0	5.68434E-14	0	9.01537E-11	0	0
	Worst	76.62989803	282.9948352	5.68434E-14	14.94139474	11.79543628	18.40187825	6.68404661	0
	Mean	54.06197808	162.7394495	3.78956E-15	2.626204088	0.546080577	9.184836132	0.39661479	0
	Std	9.810297529	43.58301536	1.44216E-14	3.749712597	2.28357397	4.675340967	1.52165200	0
F10	Best	0.002695217	0.708412743	8.88178E-16	7.54952E-14	1.15463E-14	5.21627E-12	7.99361E-	4.44089E-15
	Worst	1.360625922	19.96001708	7.99361E-15	1.46549E-13	3.64153E-14	3.58549E-11	2.22045E-	7.99361E-15
	Mean	0.185936458	14.7958024	4.79618E-15	9.64562E-14	2.19676E-14	1.64674E-11	1.5928E-14	4.79616E-15
	Std	0.445780516	7.348608771	2.696E-15	1.5843E-14	5.89582E-15	8.36597E-12	3.5751E-15	1.08403E-15
F11	Best	4.39798E-07	0.691226662	0	0	0	0	0	0
	Worst	0.039404049	90.93765307	0.216365287	0.012173631	0.039658157	0.032276084	0.01695686	0
	Mean	0.007400127	9.970233113	0.016906773	0.001148008	0.003918697	0.01048135	0.00056522	0
	Std	0.009604463	27.38383546	0.05315222	0.00351309	0.010120281	0.011535548	0.00309588	0
F12	Best	1.604334573	13619103.21	0.104111454	0.962482184	0.21421963	1.688354746	0.17418128	0.380581756
	Worst	10.35348117	325818040.9	7.096752496	5.888719656	2.945625079	12.85348765	2.38358817	1.169842467
	Mean	4.422974266	118292981.9	0.506732063	3.309574426	1.071433625	4.514060426	0.91067705	0.721041352
	Std	2.343229343	83682241.01	1.27166015	1.191318431	0.726975915	2.312196846	0.52318395	0.163739768
F13	Best	4.830503959	79928426.13	0.525800914	5.383776068	2.32031093	9.493814301	1.79363076	2.545021523
	Worst	41.92960075	585026133.6	2.676185434	26.42349085	11.53322581	37.16601697	7.33130944	3.55702194
	Mean	17.38999252	237951829.1	1.400475334	10.32643742	4.515938171	19.2946504	3.81086210	2.887597749
	Std.	9.929256026	113092504.8	0.465939596	4.177855191	1.866567399	6.642529041	1.43062099	0.193337137

In contrast to the unimodal functions, there are six multimodal functions (F8-F13) which are having a massive number of local optima. These functions are used to validate the effectiveness of the exploration performance of the WGWOIC algorithm against comparative algorithms. Table 3 lists the computational results of multimodal benchmark functions. The proposed algorithm is considerably better than all comparative algorithms on F9, F10 and F11 functions and is very competitive an the remaining multimodal functions.

Table 4. Results of fixed-dimension multimodal benchmark functions.

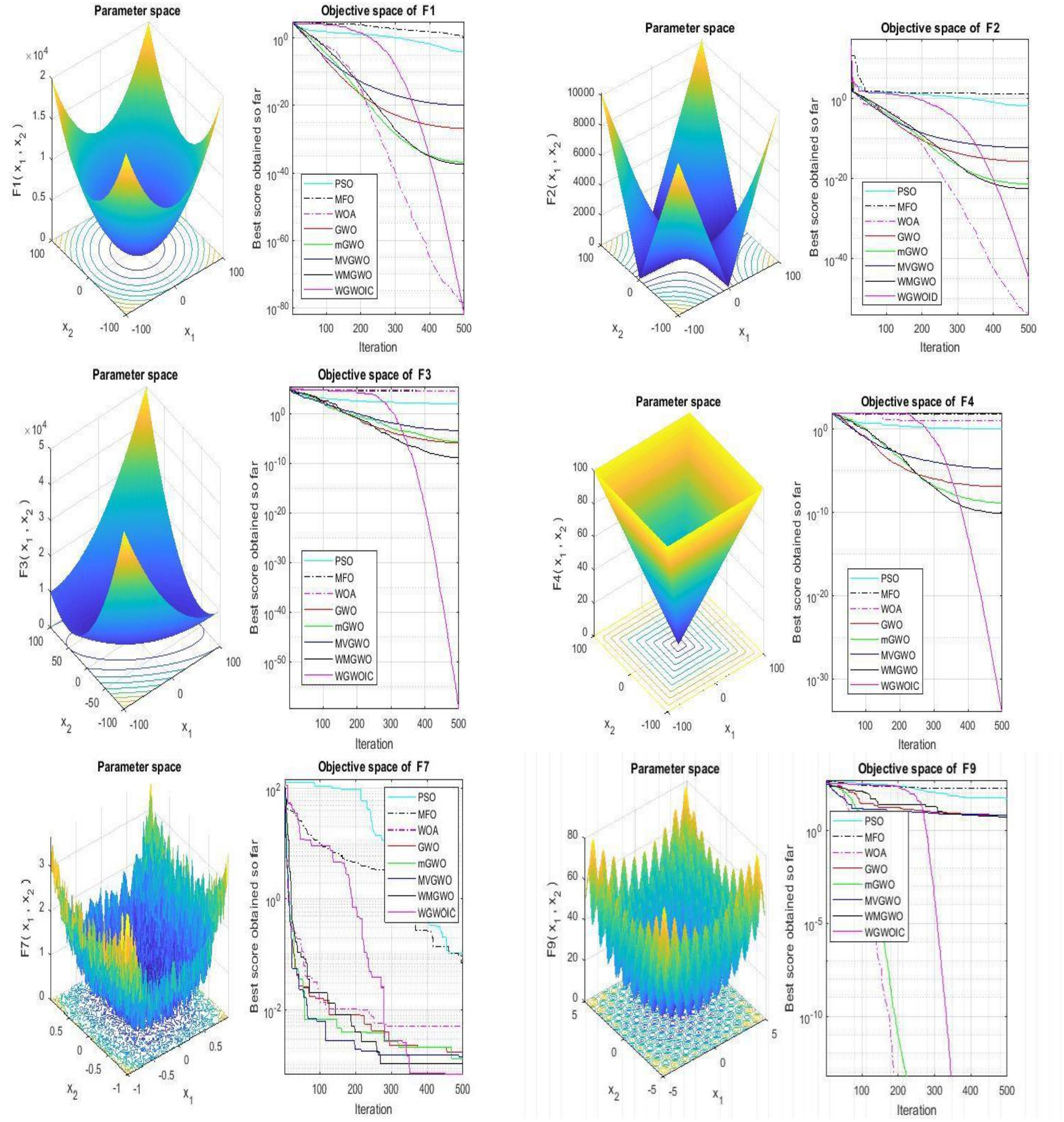
Functions	Criteria	PSO	MFO	WOA	GWO	mGWO	MVGWO	WMGWO	WGWOIC
F14	Best	0.998003838	0.998003838	0.998003838	0.998003838	0.998003838	0.998003838	0.998003838	0.998004058
	Worst	10.76318067	8.840835963	10.76318067	12.67050581	10.76318067	16.44090731	10.76318067	10.76318067
	Mean	3.55705999	2.413759985	2.408871047	4.230081157	3.093536966	6.076285786	2.442656154	2.04960213
	Std	2.963055118	1.994557984	2.53814438	4.11037804	3.210295799	4.921621581	2.458950183	2.499411434
F15	Best	0.000626681	0.000443089	0.000308353	0.000307496	0.000307649	0.000307834	0.000307991	0.000320747
	Worst	0.001594171	0.020363339	0.001606437	0.020363361	0.020363369	0.020363377	0.020363351	0.001331427
	Mean	0.000898827	0.001628901	0.000784627	0.004401317	0.005094642	0.003063687	0.001827055	0.0005669
	Std	0.000174434	0.003564441	0.000391673	0.008119157	0.008570346	0.006901842	0.005048324	0.000300483
F16	Best	-1.031628445	-1.031628453	-1.031628452	-1.031626947	-1.031628208	-1.031628131	-1.031626365	-1.031601951
	Worst	-1.02451432	-0.215463311	-0.906007779	-0.999980519	-0.999025583	-0.988939132	-0.9988901	-1.011447444
	Mean	-1.031280141	-1.004375325	-1.022052627	-1.029637815	-1.027771295	-1.028035569	-1.030414104	-1.028589411
	Std	0.001347638	0.149001867	0.028279174	0.007355832	0.009660761	0.009712765	0.005957786	0.004563169
F17	Best	0.397887358	0.397887358	0.397889389	0.3978963	0.397901342	0.397896281	0.39789264	0.398513531
	Worst	0.397903535	1.943140663	3.145500095	3.491223558	0.406921043	0.400650828	4.97391737	0.729053842
	Mean	0.397887912	0.505199511	0.691432704	0.502478431	0.400023301	0.39844679	0.557064981	0.442742179
	Std	2.95103E-06	0.391417638	0.557575662	0.564515144	0.00261759	0.00073983	0.83432747	0.065486544
F18	Best	3	3	3.000000147	3.000000205	3.000000115	3.00000016	3.000000194	3.00000061
	Worst	3	3	3.002647476	84.00001234	84.00012503	3.000177318	3.00007862	3.000237848
	Mean	3	3	3.000272395	5.700037282	5.700021785	3.000055953	3.000019692	3.000034633
	Std	1.96537E-15	2.15201E-15	0.000568143	14.78850434	14.78852855	5.11566E-05	2.03176E-05	4.77781E-05
F19	Best	-3.862782147	-3.862782148	-3.862104082	-3.862771483	-3.862606236	-3.862651908	-3.862475174	-3.862328044
	Worst	-3.862781313	-3.089764162	-2.773780759	-1.000795465	-3.810140407	-1.000783963	-3.814269526	-3.714218991
	Mean	-3.862782021	-3.824806378	-3.698618476	-3.761584254	-3.855949456	-3.761104497	-3.852313926	-3.809338982
	Std	2.39265E-07	0.152632921	0.277569519	0.521471286	0.011160136	0.521408327	0.012798558	0.03926783
F20	Best	-3.314418487	-3.06809545	-3.039630101	-3.301659308	-3.304775986	-3.288118188	-3.317169523	-3.007106385
	Worst	-0.909982661	-0.373612111	-0.43072469	-1.746400274	-1.780786493	-1.065166152	-1.769534999	-0.956158259
	Mean	-2.771059131	-2.042624029	-1.538709099	-2.862052074	-2.970472608	-2.770256553	-2.905743952	-2.132615795
	Std	0.60116875	0.867003605	0.751171691	0.424377537	0.325042763	0.546032058	0.449549744	0.667529105
F21	Best	-10.15319968	-10.15319968	-10.1508081	-10.15276796	-10.15130887	-10.15295394	-10.15211863	-4.953161739
	Worst	-2.630471668	-2.630471668	-2.627163857	-5.055188892	-2.6809166	-2.630342057	-5.054989528	-0.878075038
	Mean	-7.594555698	-6.545794	-8.674089067	-9.644703023	-9.221455628	-8.981010629	-9.130220118	-4.15674959
	Std	3.197162146	3.139790866	2.682803809	1.545931042	2.137959972	2.693260569	2.063434258	1.339198363
F22	Best	-10.40294057	-10.40294057	-10.40111333	-10.40280511	-10.40123363	-10.40284313	-10.40234302	-7.154492635
	Worst	-2.751933564	-2.751933564	-1.836472833	-1.837523021	-10.38840976	-10.39856825	-2.765762625	-2.599856113
	Mean	-9.317729361	-8.186183479	-7.787194844	-10.11578943	-10.39609553	-10.40120076	-10.14113624	-4.635777438
	Std	2.507392867	3.238901603	2.913737763	1.563515286	0.003127188	0.001018144	1.392993271	0.715724367
F23	Best	-10.53640982	-10.53640982	-10.53595185	-10.53549351	-10.53507373	-10.53612511	-10.53434444	-7.527868551
	Worst	-2.421734027	-2.421734027	-2.41789726	-2.421664232	-10.52035255	-2.421726384	-5.128106219	-0.942178182
	Mean	-9.109575858	-7.255347769	-6.655315468	-9.54281558	-10.52853524	-10.26427221	-10.16837401	-4.564613794
	Std.	2.803435329	3.659985998	3.355458636	2.607522456	0.004334215	1.481220641	1.370062816	0.991545922

On the other hand, the last ten functions (F14-F23) of the first benchmark dataset are known as fixed-dimension multimodal functions that validate the effectiveness of the exploration performance and the avoidance of local optima. Hence, the key focus is on global optima along with their exploitation performance for their convergence rate. Table 4 lists the computational results of fixed-dimension multimodal functions. The proposed algorithm outperforms all comparative well-known meta-heuristic algorithms an F14 and F15 functions and provides very competitive developments an remaining fixed-dimension multimodal functions. Hence, the proposed algorithm is validated and justified for global optimum and good convergence rate.

Table 5. Results of CEC-C06 2019 benchmark test functions.

Functions	Criteria	PSO	MFO	WOA	GWO	mGWO	MVGWO	WMGWO	WGWOIC
CEC 01	Best	84016047950	264438535.5	10875873.0	63212.9355	517904.745	678610.799	45647.22929	41835.65672
	Worst	5.74709E+12	1.13967E+11	1.34126E+1	206372812	452193546	659991656.	7947145755	40370564.89
	Mean	2.04372E+12	18693542941	3082400885	197236612	467463912.	101803359.	512481102.5	1397974.063
	Std	1.26038E+12	29133114337	3328126714	464739296.	929119650.	158517407.	1455917657	7360752.576
CEC 02	Best	8997.812591	18.99160195	17.4052621	17.3462161	17.3541055	17.3471554	17.35019871	17.45346937
	Worst	24672.5164	165.4226319	18.9502206	17.7073142	17.6953701	18.7855508	17.37192198	17.94370798
	Mean	15385.68872	52.60855145	18.0426850	17.3885923	17.3974980	17.5968274	17.3617251	17.67726882
	Std	3959.693873	30.69601113	0.41002520	0.09915725	0.09145850	0.37644568	0.005624935	0.111560162
CEC 03	Best	12.70240436	12.70240431	12.7024071	12.7024043	12.7024044	12.7024042	12.70240438	12.70242326
	Worst	12.704906	12.70254569	12.7046566	12.7047524	12.7042670	12.7059078	12.70490424	12.70310611
	Mean	12.70254677	12.70243193	12.7025459	12.7025361	12.7025086	12.7026334	12.70257894	12.7025193
	Std	0.000525338	3.65032E-05	0.00040738	0.00047321	0.00037781	0.00072414	0.000631479	0.000159984

CEC 04	Best	4.974795285	13.92941682	91.0796608	25.4822642	31.7801790	16.6507263	32.82018789	807.4234065
	Worst	45.95817019	601.633464	916.872185	1013.16275	2409.03329	2474.53275	2351.156413	4595.461272
	Mean	15.6159303	177.0484372	328.230249	95.6389720	148.620689	245.427925	152.5301426	1926.940239
	Std	8.919899821	202.1005042	175.219281	174.713382	427.382512	616.514640	416.0275936	740.7697138
CEC 05	Best	1.858024186	2.344005101	1.95590679	1.62002275	1.79651098	1.65520471	1.722212455	2.471959211
	Worst	4.812449362	4.161472658	4.12150312	2.14344097	2.10298904	2.375428153	2.325748056	3.885164162
	Mean	3.42182944	3.190601392	3.16522614	1.91598588	1.95687935	1.988312207	1.980947909	2.995941288
	Std	0.701206358	0.441984574	0.46085167	0.10467692	0.08432479	0.166134256	0.122111473	0.306795588
CEC 06	Best	5.298933027	1.165535574	7.60266370	8.98336563	9.05185916	10.02866692	6.086274462	9.75098118
	Worst	11.14441722	10.79436376	10.7467440	12.1446098	12.2400230	12.43691074	12.33565412	12.03766536
	Mean	9.055851559	5.735387803	9.43556745	11.0579200	10.8917927	11.24706021	10.93271595	10.80254285
	Std	1.375294442	2.517629806	0.96060985	0.74307568	0.79118878	0.657555238	1.215991338	0.627645061
CEC 07	Best	-74.10859072	-126.335496	208.750080	11.2883697	122.093624	-26.62196701	158.3134872	494.7379115
	Worst	374.2502338	1252.64433	1168.45006	1076.17473	1095.05153	960.270531	1074.608334	1056.631639
	Mean	150.2608467	452.6023701	660.953618	521.059954	561.461254	379.797949	558.1062293	825.942669
	Std	113.9916614	260.2282114	240.432068	290.169065	296.427841	269.355960	257.1206831	133.0243803
CEC 08	Best	3.743690545	4.33845849	4.90500812	2.98562530	2.66140208	2.62848205	3.247684807	4.818489682
	Worst	6.20863408	6.900537982	6.68769272	6.61444688	6.34279806	6.54948398	6.493956809	6.726253322
	Mean	5.147764502	5.710775289	5.81845279	4.59610904	4.52684407	4.56915741	4.852959507	5.683344627
	Std	0.56888959	0.642495043	0.48673659	0.94864667	0.96882259	0.89918493	0.998069408	0.455802888
CEC 09	Best	2.339280498	2.46217298	2.92218413	3.25728240	2.63124987	2.78707052	3.108344345	6.187666562
	Worst	2.355534157	1121.584371	6.79360902	5.71864927	5.98819892	370.750220	6.115638485	82.13447988
	Mean	2.346186889	39.94266085	4.24822211	4.33404679	4.37809308	16.4262217	4.489645472	27.96418284
	Std	0.004419399	204.2895831	0.82564146	0.83304097	0.90941748	66.9255875	0.785456542	16.20840294
CEC 10	Best	20.04350652	19.99983504	20.0762604	20.2698906	20.3293946	20.3073990	20.2193504	15.99976137
	Worst	20.64711606	20.39759247	20.6018988	20.6921739	20.6703872	20.6638055	20.62843663	20.69321035
	Mean	20.27204742	20.15907072	20.2779601	20.5085575	20.498879	20.4792108	20.50050223	20.23494306
	Std.	0.15193306	0.117342779	0.11900441	0.09854305	0.09567490	0.08456730	0.088448808	0.902719447



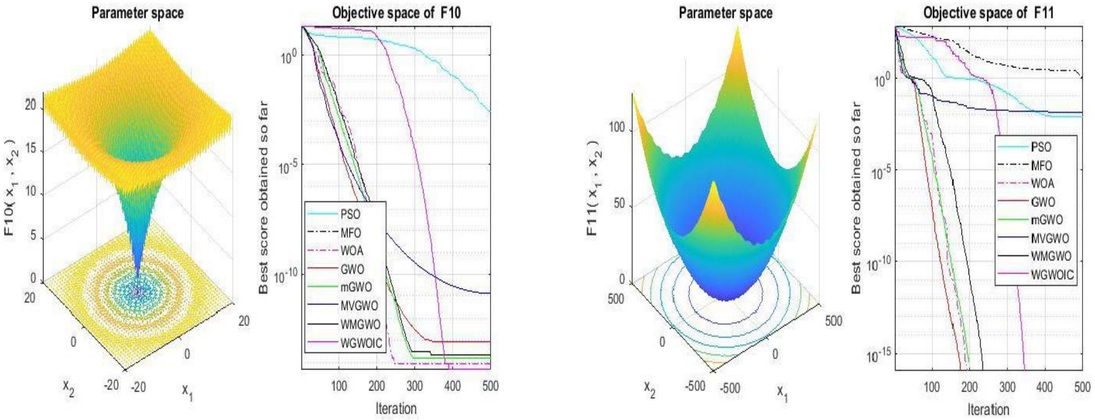


Figure 4. Convergence curve of WGWOIC algorithm and other comparative algorithms.

Subsequently, the last ten modern single objective functions (CEC01 to CEC10) have been included from CEC-C06 2019 to validate the scalability and effectiveness of the rotated and shifted ability of the proposed algorithm. Table 5 lists the computational results of these benchmark test functions. The proposed algorithm obtained very competitive effects on these functions against comparative well-known meta-heuristic algorithms. Hence, the proposed algorithm eventually offers very prominent scalability and the ability of rotated and shifted.

In addition to the above mathematical results of the experiment, we have obtained some graphical results, as depicted in Figure 4. These graphical results demonstrate the convergence rate of the proposed algorithm against other literature algorithms. The cyan color line indicates the PSO's curve graph (convergence rate graph). Similarly, the curve graphs of MFO, WOA, GWO, mGWO, MVGWO and WMGWO are shown by black color dash-dot line, magenta color dash-dot line, red color line, green color line, blue color line and black color line, respectively. In contrast, the curvegraph of WGWOIC is indicated by a magenta color line. Consequently, these graphical results justify the excellent convergence rate of the proposed algorithm compared with all comparative meta- heuristic algorithms.

5. WGWOIC IN TRAINING MULTI-LAYER PERCEPTRON

Neural Networks (NNs) [43] represents the most prominent and emerging invention in the soft computing field, proposed by McCulloch and Pitts in 1943. These networks impersonate the biological neurons of the human brain; hence they contain the ability to learn from experience. The learning methods are classified into two categories: supervised and unsupervised. As the name implies, supervised learning is provided by the supervisor or external sources (feedback). In contrast to supervised learning, unsupervised learning is conferred by merely inputs, but not accompanied by any supervisor or external sources (feedback). Neural network learning method is known as a trainer that is responsible concerning the networks' performance. Hence, the trainer is the most vital component of NNs. The set of input samples to the neural networks are notified as training samples and test samples are utilized in order to substantiate the effectiveness of their performance. However, the trainer has to accommodate the structural parameters of NNs to improve the performance in each training step. Consequently, the trainer extinctions after the training phase and the neural network are now ready to practice.

There are several types of neural networks studied in the literature, such as Recurrent Neural Networks (RNNs) [44], Feedforward Neural Networks (FNNs) [45] and so forth. The two-direction information flow between the neurons is implied in the RNNs, whereas FNNs are the simplest, most widely employed and share the information in exclusively one direction. Furthermore, FNNs are classified into two categories: Single-Layer Perceptrons (SLPs) [46] and Multi-Layer Perceptrons (MLPs) [47]. The SLPs comprise only one perceptron that constitutes it suitable to solve linear problems. In contrast to SLPs, MLPs consist of more than one perceptron at several layers, making them ideal for solving non-linear problems. There are proposed numerous applications of MLPs in the literature, such as function approximation, pattern classification and so forth. The pattern classification [48] is a supervised learning approach and it classifies the input data in to preconcert labeled classes, whereas

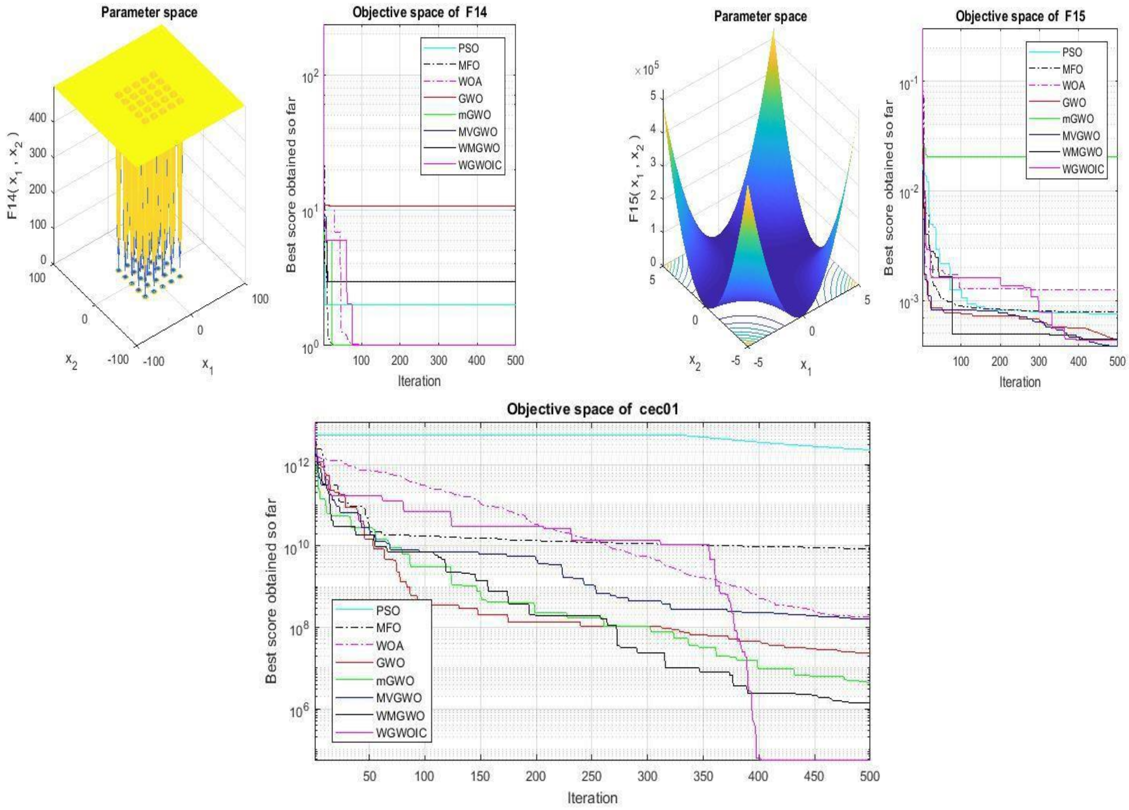


Figure 4. (*continued*).

function approximation [49] concerns the undertaking of modeling relationships amid input variables. The hierarchical classification diagram of neural networks is depicted in Figure 5.

Yu et al. [50] introduced an algorithm in order to train Support Vector Machines (SVMs) to alleviate computational complexities. The proposed algorithm utilized Fisher projection for bound vectors set to tackle linear and non-liner separates problems, for which linear and kernel Fisher discriminants were used to compute projection line. Yu et al. [51] proposed a two-side (user-side and item-side) Cross Domain Collaborate Filtering (CDCF) algorithm in order to diminish the sparsity problem that occurred in the recommender systems. The recommendation problem is converted into a classification problem by using the proposed model, alongside the SVM model employed to tackle the resultant classification problem that eventually performs significantly better than comparative methods.

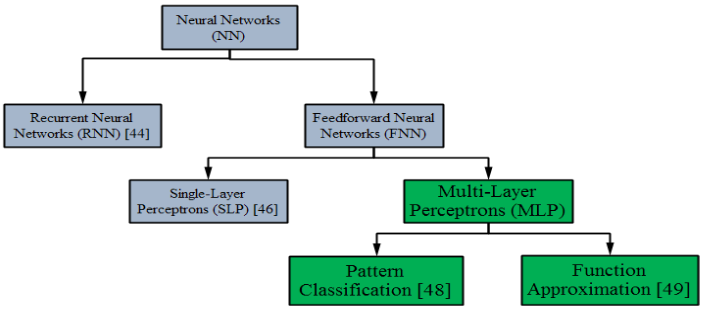

Neural Networks (NN)
Recurrent Neural Networks (RNN) [44]
Feedforward Neural Networks (FNN)
Single-Layer Perceptrons (SLP) [46]
Multi-Layer Perceptrons (MLP)
Pattern Classification [48]
Function Approximation [49]

Figure 5. Hierarchical classification of neural networks.

Furthermore, Yu et al. [52] enhanced the CDCF algorithm and expanded user and item (two-dimensional) location as the feature vector using the Funk-SVD decomposition model. Further, the C4.5 decision tree algorithm has been utilized for training a classifier for predicting missing ratings.

Kolisetty and Rajput [53] contributed to intensive study regarding the significance of machine learning in analyzing big data's analysis (implications and challenges) in terms of data heterogeneity, classification imperfection and computational complexity. The analyzed data is utilized for predictive analysis and decision-making *via* data transformation and knowledge extraction. Furthermore, Ottom's critical focus [54] has emphasized the significance of big data in healthcare and its implementation tools and associated research challenges. The effectiveness of the Fuzzy-Rough Nearest Neighbour (FRNN) classifier is benchmarked by Liew et al. [55] for brainprint authentication over nine distinctive electroencephalogram channels' signals. Nahar et al. [56] employed a user-defined lexicon approach in order to determine the polarity (positive, negative) on Facebook posts and comments and achieved 98% accuracy. Apart from the above method, Naïve Bayes (NB), K-Nearest Neighbour (K-NN) and support vector machine classifiers have been utilized for polarity classification and produced 95.6, 96.8 and 97.8%, respectively. Al-Abdallah et al. [57] proposed a firefly algorithm classifier in order to tackle five binary classification problems. The effectiveness of the proposed classifier demonstrates competitive outcome against comparative classifiers. Alweshah et al. [58] proposed a hybrid approach (African buffalo optimization algorithm employed with the probabilistic neural network) to address the classification problem and applied it on 11 benchmark datasets in order to assess its accuracy. Furthermore, the water evaporation algorithm employed with the probabilistic neural network in order to tackle classification problems very effectively was developed by Alweshah et al. [59].

As we discussed earlier, the trainer is the most influential component of neural networks. There are recommended several trainers to NNs in the literature, such as Genetic Algorithms (GAs) [60], Particle Swarm Optimization (PSO) [61], Evolutionary Strategies (ESs) [62], Ant Colony Optimization (ACO) [63]-[64], Grey Wolf Optimizer (GWO) [65]-[66], Teaching-Learning Based Optimization (TLBO) [67], Moth-Flame Optimizer (MFO) [68]-[69], Population-based Incremental Learning (PBIL) [70] and so forth. In order to find a most optimal prediction for Dairy Product Demand (DPD) in Iran, Goli et al. [71] proposed a hybrid approach using GWO and cultural algorithm to improve MLPs. We are focusing on tackling the pattern classification problem using MLPs in this research work employed by the preceding proposed novel algorithm of GWO designated as WGWOIC trainer. The proposed trainer may assist in this field and provide considerably better outcomes than other comparative trainers.

5.1 Problem Formation

As discussed earlier, MLPs are specific types of Feedforward Neural Networks that contain one hidden layer with one input and one output layer. However, the output of MLPs depends on the inputs, weights and biases. In order to tackle the pattern classification datasets, these datasets already contain the inputs and outputs, while optimum weights and biases are also required for significant computational results. In this research work, we proposed an algorithm-based trainer called Weighted Grey Wolf Optimizer with Improved Convergence Rate-Multi-Layer Perceptron trainer (WGWOIC-MLP trainer) in order to optimize the values of weights and biases. The block diagram of this proposed work regarding problem formulation is depicted in Figure 6.

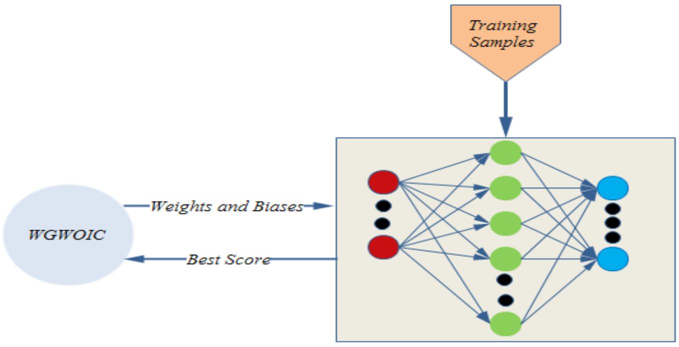


Figure 6. Block diagram of the proposed work.

The block diagram illustrates that the proposed algorithm trainer provides optimized weights and biases to the MLP that returns the best score for the testing samples. The performance is benchmarked on 3-bits XOR, balloon, iris, breast cancer and heart datasets that are well-known classification datasets in the literature. These datasets are of different difficulty levels that are considered from University of California at Irvine (UCI) Machine Learning Repository and the researchers may follow reference [61] for a detailed description of them. Therefore, the number of attributes is represented by

#Attributes for each dataset according to Table 6. Similarly, the number of training samples (#Training Samples) and test samples (#Testing Samples), number of classes (#Classes) and the corresponding MLP structure for each dataset are listed in Table 6.

Table 6. Details of pattern classification datasets.

Classification Datasets	#Attributes	#Training Samples	#Testing Samples	#Classes	MLP Structure
3-bits XOR	3	8	8	2	3-7-1
Balloon	4	16	16	2	4-9-1
Iris	4	150	150	3	4-9-3
Breast Cancer	9	599	100	2	9-19-1
Heart	22	80	187	2	22-45-1

The number of nodes in each MLP structure's hidden layer is two times the number of inputs plus one more (2*N+1, where N indicates the number of inputs in the particular dataset). The simulation results and discussion are described in the following sub-section. It may be noted that the training algorithms are represented by algorithm-MLP in this simulation.

5.2 Simulation Results and Discussion

The programming implementation for this research application has been done on the same platform as the preceding experiments. However, the settings of tuning parameters for the WGWOIC and other state-of-the-art meta-heuristic trainers are listed in Table 7. The population size (search agents) for each MLP training algorithm is 50 for the XOR and Balloon datasets, while it is 200 for the rest datasets. At the same time, the maximum number of generations is 250 for each training algorithm.

The classification rate and best score have been considered as performance-evaluating parameters of these training algorithms. However, the highest value of classification rate and lowest value of the best score indicate the most optimal solution. The best score is also called mean square error that is calculated by the difference between the actual value and the desired value of the individual sample.

According to Table 6, the XOR dataset contains three attributes, eight training/test samples and two classes. The MLP structure of this dataset is 3-7-1, implying that the multi-layer perceptron neural network contains three inputs nodes, seven hidden nodes and one output node and the trainer has 36 dimensions. The results of sub-experiments of all datasets are depicted in Figure 7 and listed in Table 8, which demonstrate that WGWOIC-MLP and GA-MLP trainers provide a 100 percent classification rate (accuracy) to classifying the XOR dataset. In contrast, the MFO-MLP trainer

Table 7. The initial parameters of training algorithms.

Training Algorithm	Parameter	Value
WGWOIC	$\vec{a}$	Exponentially decrease from 2 to 0
	Population	50 for the XOR and Balloon, 200 for rest
	#Generation	250
GA	Crossover	Single point (probability=1.0)
	Mutation	Uniform (Probability=0.01)
	Type	Real Coded
PSO	Topology	Fully Connected
	Social constant (*C2*)	1
	Cognitive constant (*C1*)	1
	Inertia constant (*w*)	0.3
ACO	Initial pheromone (τ)	1e-06
	Pheromone update constant (*Q*)	20
	Pheromone constant (*q*)	1
	Global pheromone decay rate (p_g)	0.9
	Local pheromone decay rate (p_t)	0.5
	Pheromone sensitivity (α)	1
	Visibility sensitivity (β)	5
ES	λ	10
	σ	1
MFO	*b*	1
	t	[-1, 1]
PBIL	Mutational probability	0.1
	Learning rate	0.05
	Elitism parameter	1

provides the highest best score against other well-known comparative trainers, whereas GA and WGWOIC-MLP also offer very significant results. In the case of the balloon dataset, it possesses four features, 16 training/test samples and two classes, according to Table 6. The MLP structure of the current dataset is 4-9-1 and the trainer has to optimize 55 variables. Surprisingly, all trainers' classification rates are 100 percent, whereas the GA-MLP achieves the best score while the WGWOIC-MLP trainer reached the second rank.

The iris is the most popular dataset in the literature and consists of four attributes, 150 training/test samples and three classes. The MLP structure of this current dataset is 4-9-3 and the trainer has 75 dimensions that have to be optimized. The experimental results clarify that the MFO-MLP trainer provides the highest classification rate, while WGWOIC-MLP trainer obtained the second-best accuracy. Moreover, the WGWOIC-MLP trainer offers the highest best score compared to the other trainers.

The breast cancer dataset is also a well-known dataset in the literature and contains nine features, 599 training samples, 100 test samples and two classes. The MLP structure of this dataset is 9-19-1 and the trainer has 209 variables. The experimental results show that both WGWOIC and MFO-MLP trainers provide 99 percent accuracy, while GA offers 98 percent classification rate. Moreover, the MFO-MLP trainer offers the highest best score among other meta-heuristic comparative trainers, while the WGWOIC-MLP trainer acquired the second rank.

The heart dataset is the most challenging dataset in the studied literature and consists of 22 attributes, 80 training samples, 187 test samples and two classes. The MLP structure of this current dataset is 22- 45-1 and the trainer has to optimize 1081 variables. The experimental results demonstrate that the WGWOIC-MLP trainer acquired the highest classification rate, while the MFO-MLP trainer provided the second-highest accuracy. Moreover, the GA and WGWOIC-MLP based trainer provides the highest best score compared other trainers.

To sum up, the experimental results justify that the proposed WGWOIC-MLP trainer in order to tackle the pattern classification problems provides significantly better outcomes than other well-known comparative trainers. In addition, the proposed algorithm extensively examines the search space in order to avoid the local optima and simultaneously offers an excellent equilibrium among exploration and exploitation to tackle the optimization problems; thus, its promising exploitation devotes considerably better convergence rate toward the most optimal solution to the proposed algorithm.

Table 8. Best score for the XOR, Balloon, Iris, Breast cancer and Heart datasets.

Training Algorithm	**Pattern Classification Datasets**				
	XOR (Best Score)	Balloon (Best Score)	Iris (Best Score)	Breast Cancer (Best Score)	Heart (Best Score)
WGWOIC	0.0057382	3.8195E-22	0.016928	0.0021115	0.129182
MFO	0.0000454	1.0876E-20	0.021996	0.0019964	0.178128
PSO	0.0750883	2.9306E-05	0.134318	0.0267692	0.159182
PBIL	0.0262799	5.2337E-06	0.059117	0.0243935	0.135457
GA	0.0002162	1.2192E-24	0.022447	0.0026742	0.075491
ES	0.1057665	0.0023197	0.299674	0.0401639	0.169544
ACO	0.1172278	0.0017509	0.327935	0.0114927	0.219699

In addition to the classification problem, the proposed method may be employed to tackle several potential applications of various crucial research domains of science and technology, such as machine learning applications (Training neural networks, feature selection, data clustering, optimizing SVMs), image processing applications (image thresholding, image classification), wireless sensor network applications (extending the network lifetime, network coverage problem, localization problem), engineering applications (robotics and path planning, power dispatch problems, designing and tuning controllers), Controller Placement Problem (CPP) in software-defined networking, software cost estimation and so forth.

The proposed method provides considerably better performance in terms of exploration and exploitation of the search space. The finding of this research work is that the outcomes of the WGWOIC algorithm are significantly better on high-dimensional functions, whereas it lacks in terms

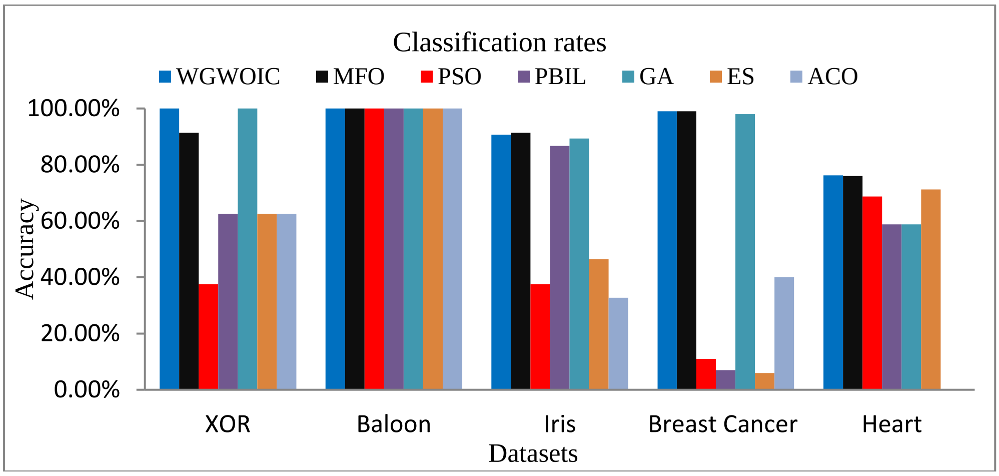


Figure 7. Classification rates for the XOR, Balloon, Iris, Breast cancer and Heart datasets.

of low-dimensional functions. However, the result analysis elaborates that the proposed method offers comparatively better results if it is employed with less population and fewer iterations to optimize low-dimensional problems. Moreover, there is scope for further research to overcome the above limitation adding some new operators or modifying the existing ones. On the other hand, the proposed method considerably tackles the classification problem of the real world.

6. CONCLUSION

This research work intends to introduce a novel algorithm of grey wolf optimizer to overcome its major impediment (stagnation in local optima) and the limitations of other algorithms. Further, the suggested algorithm is employed being a trainer of MLP neural networks to improve the accuracy of the classification problem. This paper introduced the novel algorithm nominated Weighted Grey Wolf Optimizer with Improved Convergence Rate (WGWOIC) for extensive exploration of the search space. Therefore, this research work has enhanced the hunting (position update equation) and the attacking (exploitation equation) mechanisms of basic GWO. In order to test the effectiveness of the WGWOIC algorithm's performance, it is benchmarked on 33 fairly popular numerical test functions that are considered from two different benchmark datasets. The experimental results of the benchmark datasets assist in justifying the strength and robustness of the proposed algorithm against the unknown search space of real-world applications. Surprisingly, the recommended algorithm outperforms on the majority of test functions against comparative studied meta-heuristic optimization algorithms, whereas it provides very competitive results on the remaining functions.

In addition, the WGWOIC algorithm was further employed as a trainer for multi-layer perceptron to classify five viral pattern classification datasets. Conclusively, it produces very competitive outcomes regarding classification rate and best score, demonstrating that the proposed algorithm is robust against challenging problems with unknown search spaces.

The experimental finding of the WGWOIC algorithm is that the proposed algorithm discovers better-quality solutions in terms of high exploration and exploitation abilities of abstract search space of numerical and real-world problems. The proposed algorithm may be further employed to tackle several potential applications of various crucial research domains; for instance, machine learning, image processing, wireless sensor network applications, controller placement problem in SDN and so forth. In addition, this algorithm may further improve after adopting the evolutionary mechanism, which is worth being the subject of further research works.

REFERENCES

[1] D. Wolpert and W. Macready, "No Free Lunch Theorems for Optimization," IEEE Trans. on Evolutionary Computation, vol. 1, no. 1, pp. 67-82, April 1997.

[2] J. Holland, "Genetic Algorithms," Scientific American, vol. 267, no. 1, pp. 66-73, July 1992.

[3] K. Krishnakumar and D. E. Goldberg, "Control System Optimization Using Genetic Algorithms," Journal of Guidance, Control and Dynamics, vol. 15, no. 3, pp. 735-740, 1992.

[4] R. Storn and K. Price, "Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," J. of Global Optimization, vol. 11, no. 4, pp. 341-359, 1997.

[5] D. Simon, "Biogeography-based Optimization," IEEE Trans. on Evolutionary Computation, vol. 12, no. 6, pp. 702-713, March 2008.

[6] X. Yao, Y. Liu and G. Lin, "Evolutionary Programming Made Faster," IEEE Trans. on Evolutionary Computation, Vol. 3, no. 2, pp. 82-102, July 1999.

[7] J. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, vol. 1, MIT Press, 1992.

[8] N. Hansen, S. Müller and P. Koumoutsakos, "Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES)," Evolutionary Computation, Vol. 11, no. 1, pp. 1- 18, March 2003.

[9] X. Yao and Y. Liu, "Fast Evolutionary Programming," Evolutionary Programming, vol. 3, pp. 451-460, Feb. 1996.

[10] S. Hofmeyr and S. Forrest, "Architecture for an Artificial Immune System," Evolutionary Computation, vol. 8, no. 4, pp. 443-473, December 2000.

[11] K. Passino, "Bacterial Foraging Optimization," International Journal of Swarm Intelligence Research (IJSIR), vol. 1, no. 1, pp. 1-16, January 2010.

[12] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, "GSA: A Gravitational Search Algorithm," Information Sciences, vol. 179, no. 13, pp. 2232-2248, June 2009.

[13] B. Webster and P. Bernhard, "A Local Search Optimization Algorithm Based on Natural Principles of Gravitation," Proc. of the 2003 International Conf. on Information and Knowledge Engineering (IKE'03), pp. 255–261, Las Vegas, Nevada, USA, April 2003.

[14] A. Hatamlou, "Black Hole: A New Heuristic Optimization Approach for Data Clustering," Information Sciences, vol. 222, pp. 175-184, February 2013.

[15] F. Moghaddam, R. Moghaddam and M. Cheriet, "Curved Space Optimization: A Random Search Based on General Relativity Theory," arXiv Preprint arXiv: 1208.2214, August 2012.

[16] X. Yang, "A New Metaheuristic Bat-inspired Algorithm," Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Part of Studies in Computational Intelligence Book Series, vol. 284, pp. 65-74, Springer, Berlin, Heidelberg, 2010.

[17] S. Mirjalili, "Moth-flame Optimization Algorithm: A Novel Nature-inspired Heuristic Paradigm," Knowledge- based Systems, vol. 89, pp. 228-249, November 2015.

[18] B. Mohanty, "Performance Analysis of Moth Flame Optimization Algorithm for AGC System," International Journal of Modeling and Simulation, vol. 39, no. 2, pp. 73-87, April 2019.

[19] D. Pelusi, R. Mascella, L. Tallini, J. Nayak et al., "An Improved Moth-flame Optimization Algorithm with Hybrid Search Phase," Knowledge-based Systems, vol. 191, ID: 105277, 2020.

[20] P. Singh and SK. Bishnoi, "Modified Moth-flame Optimization for Strategic Integration of Fuel Cell in Renewable Active Distribution Network," Electric Power Systems Research, vol. 197, Article ID: 107323, 2021.

[21] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," Advances in Engineering Software, vol. 95, pp. 51-67, May 2016.

[22] B. H. Abed-alguni, "Bat Q-learning Algorithm," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 03, no. 01, pp. 52-71, DOI: 10.5455/jjcit.71-1480540385, April 2017.

[23] E. Cuevas, A. Echavarría and M. Ramírez-Ortegón, "An Optimization Algorithm Inspired by the States of Matter that Improves the Balance between Exploration and Exploitation," Applied Intelligence, vol. 40, no. 2, pp. 256-272, March 2014.

[24] X. Yang, "Flower Pollination Algorithm for Global Optimization," Proc. of International Conf. on Unconventional Computing and Natural Computation (UCNC 2012), Part of the Lecture Notes in Computer Science Book Series, vol. 7445, pp. 240-249, Springer, Berlin, Heidelberg, September 2012.

[25] A. G. Hussien, D. Oliva, E. Houssein, A. Juan and X. Yu, "Binary Whale Optimization Algorithm for Dimensionality Reduction," Mathematics, vol. 8, no. 10, 1821, October 2020.

[26] AK. Tripathi, H. Mittal, P. Saxena and S. Gupta, "A New Recommendation System Using Map-reduce-based Tournament Empowered Whale Optimization Algorithm," Complex & Intelligent Systems, vol. 7, no. 1, pp. 297-309, February 2021.

[27] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," Proceedings of the IEEE International Conference on Neural Networks (ICNN'95), vol. 4, pp. 1942-1948, November 1995.

[28] K. Parsopoulos and M. Vrahatis, "UPSO: A Unified Particle Swarm Optimization Scheme," Proc. of the International Conference of Computational Methods in Sciences and Engineering (ICCMSE 2004), CRC Press, pp. 868-873, April 2019.

[29] A. Piotrowski, J. Napiorkowski and A. E. Piotrowska, "Population Size in Particle Swarm Optimization," Swarm and Evolutionary Computation, vol. 58, Article ID: 100718, November 2020.

[30] N. S. Alshdaifat and M. H. Bataineh, "Optimizing and Thinning Planar Array Using Chebyshev Distribution and Improved Particle Swarm Optimization," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 01, no. 01, pp. 31-41, December 2015.

[31] S. Parsons, "Ant Colony Optimization by Marco Dorigo and Thomas Stützle, MIT Press, ISBN 0-262-04219-3," The Knowledge Engineering Review, vol. 20, no. 1, pp. 92, 2005.

[32] XS. Yang, "Firefly Algorithm, Stochastic Test Functions and Design Optimization," International Journal of Bio-inspired Computation, vol. 2, no. 2, pp. 78-84, January 2010.

[33] X. Yang and S. Deb, "Cuckoo Search *via* Lévy Flights," Proc. of IEEE 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 210-214, Coimbatore, India, December 2009.

[34] S. Mirjalili, S. Mirjalili S and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46-61, March 2014.

[35] N. Mittal, U. Singh and B. Sohi, "Modified Grey Wolf Optimizer for Global Engineering Optimization," Applied Computational Intelligence and Soft Computing, vol. 2016, Article ID: 7950348, March 2016.

[36] N. Singh, "A Modified Variant of Grey Wolf Optimizer," International Journal of Science & Technology, Scientia Iranica, DOI: 10.24200/SCI.2018.50122.1523, 2018.

[37] N. Singh and S. Singh, "A Modified Mean Gray Wolf Optimization Approach for Benchmark and Biomedical Problems," Evolutionary Bioinformatics, vol. 13, DOI: 10.1177/1176934317729413, 2017.

[38] A. Kumar, A. Singh and A. Kumar, "Weighted Mean Variant with Exponential Decay Function of Grey Wolf Optimizer on Applications of Classification and Function Approximation Dataset," Proc. of the International Conference on Hybrid Intelligent Systems, Springer, Cham, pp. 277-290, December 2019.

[39] B. H. Abed-alguni and M. Barhoush, "Distributed Grey Wolf Optimizer for Numerical Optimization Problems," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 04, no. 03, pp. 1-20, DOI: 10.5455/jjcit.71-1532897697, December 2018.

[40] K. Price, N. Awad, M. Ali and P. Suganthan, "The 100-digit Challenge: Problem Definitions and Evaluation Criteria for the 100-digit Challenge Special Session and Competition on Single Objective Numerical Optimization," Technical Report, Nanyang Technological University, November 2018.

[41] M. Abdullah and T. Ahmed, "Fitness Dependent Optimizer Inspired by the Bee Swarming Reproductive Process," IEEE Access, vol. 7, pp. 43473-43486, March 2019.

[42] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 2nd Edn., Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2, pp. 111–114, 2003.

[43] W. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," The Bulletin of Mathematical Biophysics, vol. 5, no. 4, pp. 115-133, December 1943.

[44] G. Dorffner, "Neural Networks for Time Series Processing," Neural Network World, vol. 6, pp.447-468, 1996.

[45] G. Bebis and M. Georgiopoulos, "Feed-forward Neural Networks," IEEE Potentials, vol. 13, no. 4, pp. 27- 31, October 1994.

[46] P. Auer, H. Burgsteiner and W. Maass, "A Learning Rule for Very Simple Universal Approximators Consisting of a Single Layer of Perceptrons," Neural Networks, vol. 21, no. 5, pp. 786-795, June 2008.

[47] P. Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. Dissertation, Harvard University, 1974.

[48] P. Melin, D. Sánchez and O. Castillo, "Genetic Optimization of Modular Neural Networks with Fuzzy Response Integration for Human Recognition," Information Sciences, vol. 197, pp. 1-19, August 2012.

[49] W. Gardner and S. Dorling, "Artificial Neural Networks (the Multilayer Perceptron): A Review of Applications in the Atmospheric Sciences," Atmospheric Environment, vol. 32, no. (14-15), pp. 2627-2636, August 1998.

[50] X. Yu, J. Yang and Z. Xie, "Training SVMs on a Bound Vectors Set Based on Fisher Projection," Frontiers of Computer Science, vol. 8, no. 5, pp. 793-806, October 2014.

[51] X. Yu, Y. Chu, F. Jiang, Y. Guo and D. Gong, "SVMs Classification Based Two-side Cross Domain Collaborative Filtering by Inferring Intrinsic User and Item Features," Knowledge-based Systems, vol. 141, pp. 80-91, February 2018.

[52] X. Yu, F. Jiang, J. Du and D. Gong, "A Cross-domain Collaborative Filtering Algorithm with Expanding User and Item Features *via* the Latent Factor Space of Auxiliary Domains," Pattern Recognition, vol. 94, pp. 96-109, October 2019.

[53] V. V. Kolisetty and D. S. Rajput, "A Review on the Significance of Machine Learning for Data Analysis in Big Data," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 01, pp. 155- 171, DOI: 10.5455/jjcit.71-1564729835, March 2020.

[54] M. A. Ottom, "Big Data in Healthcare: Review and Open Research Issues," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 03, no. 01, pp. 38-51, DOI: 10.5455/jjcit.71-1476816159, April 2017.

[55] S.-H. Liew, Y.-H. Choo and Y. F. Low, "Fuzzy-rough Classification for Brainprint Authentication," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 05, no. 02, pp. 52-71, DOI: 10.5455/jjcit.71-1556703387, August 2019.

[56] K. M.O. Nahar, A. Jaradat, M. S. Atoum and F. Ibrahim, "Sentiment Analysis and Classification of Arab Jordanian Facebook Comments for Jordanian Telecom Companies Using Lexicon-based Approach and Machine Learning," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 03, pp. 52-71, DOI: 10.5455/jjcit.71-1586289399, Sep. 2020.

[57] R. Z. Al-Abdallah, A. S. Jaradat, I. Abu Doush and Y. A. Jaradat, "A Binary Classifier Based on Firefly Algorithm," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 03, no. 03, pp. 32- 46, DOI: 10.5455/jjcit.71-1501152301, December 2017.

[58] M. Alweshah, L. Rababa, M. H. Ryalat, A. Al Momani and M. F. Ababneh, "African Buffalo Algorithm: Training the Probabilistic Neural Network to Solve Classification Problems," Journal of King Saud University - Computer and Information Sciences, DOI: 10.1016/j.jksuci.2020.07.004, 2020.

[59] M. Alweshah, E. Ramadan, M. H. Ryalat, M. Almi'ani and A. I. Hammouri, "Water Evaporation Algorithm with Probabilistic Neural Network for Solving Classification Problems," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 1, pp. 1-14, March 2020.

[60] S. Tang, M. Li, F. Wang, Y. He and W. Tao, "Fouling Potential Prediction and Multi-objective Optimization of a Flue Gas Heat Exchanger Using Neural Networks and Genetic Algorithms," International Journal of Heat and Mass Transfer, vol. 152, Article ID: 119488, May 2020.

[61] M. F. Ab Aziz, S. A. Mostafa, C. F. M. Foozy, M. A. Mohammed, M. Elhoseny and A. Z. Abualkishik, "Integrating Elman Recurrent Neural Network with Particle Swarm Optimization Algorithms for an Improved Hybrid Training of Multidisciplinary Datasets," Expert Systems with Applications, vol. 183, p. 115441, June 2021.

[62] F. E. Fernandes Jr and G. G. Yen, "Pruning of Generative Adversarial Neural Networks for Medical Imaging Diagnostics with Evolution Strategy," Information Sciences, vol. 558, pp. 91-102, May 2021.

[63] A. Zannou and A. Boulaalam, "Relevant Node Discovery and Selection Approach for the Internet of Things Based on Neural Networks and Ant Colony Optimization," Pervasive and Mobile Computing, vol. 70, Article ID: 101311, January 2021.

[64] H. Zhang, H. Nguyen, X. Bui et al., "Developing a Novel Artificial Intelligence Model to Estimate the Capital Cost of Mining Projects Using Deep Neural Network-based Ant Colony Optimization Algorithm," Resources Policy, vol. 66, Article ID: 101604, June 2020.

[65] S. Mirjalili, "How Effective Is the Grey Wolf Optimizer in Training Multi-layer Perceptrons?" Applied Intelligence, vol. 43, no. 1, pp. 150-161, July 2015.

[66] H. Faris, S. Mirjalili and I. Aljarah, "Automatic Selection of Hidden Neurons and Weights in Neural Networks Using Ggrey Wolf Optimizer Based on a Hybrid Encoding Scheme," International Journal of Machine Learning and Cybernetics, vol. 10, no. 10, pp. 2901-2920, October 2019.

[67] E. Uzlu, M. Kankal, A. Akpınar and T. Dede, "Estimates of Energy Consumption in Turkey Using Neural Networks with the Teaching–learning-based Optimization Algorithm," Energy, vol. 75, pp. 295-303, October 2014.

[68] W. Yamany, M. Fawzy, A. Tharwat and A. Hassanien, "Moth-flame Optimization for Training Multi-layer Perceptrons," Proc. of the 11th IEEE International Computer Engineering Conference (ICENCO), pp. 267-272, Cairo, Egypt, December 2015.

[69] R. Singh, S. Gangwar, D. Singh and V. Pathak, "A Novel Hybridization of Artificial Neural Network and Moth-flame Optimization (ANN–MFO) for Multi-objective Optimization in Magnetic Abrasive Finishing of Aluminium 6060," Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol. 41, no. 6, pp. 1-19, June 2019.

[70] R. Vasco-Carofilis, M. Gutiérrez-Naranjo and M. Cárdenas-Montes, "PBIL for Optimizing Hyperparameters of Convolutional Neural Networks and STL Decomposition," Proc. of the International Conference on Hybrid Artificial Intelligence Systems, Springer, Cham, pp. 147-159, DOI: 10.1007/978-3-030-61705-9_13, November 2020.

[71] A. Goli, H. K. Zare, R. T. Moghaddam and A. Sadeghieh, "An Improved Artificial Intelligence Based on Gray Wolf Optimization and Cultural Algorithm to Predict Demand for Dairy Products: A Case Study," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, no. 6, pp. 15-22, March 2019.

ملخص البحث:

تعمل هذه الورقة على تحسين آلية الصيد والهجوم من أجل تعديل معادلة الموقع المحدَّثة ومعادلة الاستكشاف –على الترتيب- لاقتراح خوارزمية مبتكرة تحمل اسم "نظام إيجاد الحلول المُثلى باستخدام طريقة الذّئاب الرّمادية الموزونة مع معدّل التقاء محسّن". وقد تمّ استقصاء فاعلية الخوارزمية المقترحة باختبارها على (33) وظيفة مرجعيّة مختلفة شائعة الى حدّ ما. وقد اختبرت تلك الوظائف المرجعيّة من مجموعتي بياناتٍ مختلفتين لتقييم قوّة الخوارزمية المقترحة ومتانتها فيما يتعلّق بحيّز البحث غير المعروف للمشكلة.

وبهدف تحليل الأداء، تمت مقارنة نتائج الخوارزمية المقترحة مع نتائج خوارزميات أخرى جرى التّطرُّق إليها في أدبيات الموضوع، مثل خوارزميات إيجاد الحلول المُثلى القائمة على استراتيجية كلّ من أسراب الدّقائق، وعثّة النّار، والحيتان، والذّئاب الرّمادية، وأحدث الخوارزميات المستندة الى استراتيجية الذّئاب الرّمادية. وتوضح نتائج المقارنة أنّ الخوارزمية المقترحة أبْلَتْ بلاءً حسناً مقارنةً بغيرها من الخوارزميات؛ فقد تفوّقت عليها في عددٍ من الوظائف وكانت منافسةً لها في وظائف أخرى.

من ناحية أخرى، تم تهجين الخوارزمية المقترحة بشبكة عصبيّة متعدّدة الطّبقات لتحسين الدّقة فيما يتعلّق بمشكلات التّصنيف؛ إذ يوفر "مدرّب" الخوارزمية المقترحة القيم المُثلى للوزن والانحيازللشبكة العصبيّة متعدّدة الطّبقات. كذلك تمّ فحص الأداء من حيث دقّة التّصنيف باستخدام خمس مجموعات بياناتٍ شائعة، وتقييم فاعليّة "مدرّب" النّظام المقترح بالمقارنة مع أنظمةٍ أخرى مشابهة. وأشارت النتائج الى أنّ الخوارزمية المقترحة كانت ذات تنافسيّة جيّدة من حيث استغلال واستكشاف حيّز البحث وحلّ العديد من مشكلات التّصنيف بفاعلية.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 03, September 2021.

COMPARATIVE STUDY OF MACHINE LEARNING AND DEEP LEARNING ALGORITHM FOR FACE RECOGNITION

Nikita Singhal[1], Vaishali Ganganwar[2], Menka Yadav[3], Asha Chauhan[4], Mahender Jakhar[5] and Kareena Sharma[6]

(Received: 28-Jun.-2021, Revised: 17-Aug.-2021, Accepted: 19-Aug.-2021)

ABSTRACT

In the present world, biometric systems are used to analyze and verify a person's distinctive bodily or behavioral features for authentication or recognition. Till now, there are numerous authentication systems that use iris, fingerprint and face feature for identification and verification, where the face recognition-based systems are most widely preferred, as they do not require user help every time, are more automated and are easy to function. This review paper provides a comparative study between various face recognition techniques and their hybrid combinations. The most commonly used datasets in this domain are also analyzed and reviewed. We have also highlighted the future scope and challenges in this domain, as well as various Deep Learning (DL)-based algorithms for facial recognition.

KEYWORDS

Face recognition, Local binary pattern, Convolutional neural networks, Principal component analysis, Histogram of oriented gradient.

1. INTRODUCTION

With the evolution of humans in every field of technology, there is a need to control who can access the place, machinery or information; so, we require an authentication system. There are many human authentication systems, such as signature, password, pin and biometric systems that have been developed. Face authentication systems have become popular as they doesn't disturb the privacy of the individual and there is no requirement to get in physical contact with the system, which helps in controlling the spread of diseases like viruses. Face authentication is defined as giving access to the authorized person; i.e., face identification problem. It is a two-step process; firstly face detection, which is the detection of the human face in the frame of the image or video and highlighting it by making a square around the face discarding the surrounding and secondly Face Recognition (FR), which means the face detected in the above step has to be verified with those present in the database and if there exists a match, then the person is authorized by the system; if not, then the owner can take the necessary measures. There are many factors the affect the FR algorithm, including physical factors (e.g. illumination, occlusion) as well as facial features (e.g. twins, relatives, pose and aging factor). The methods addressing all these issues have been surveyed in [1] by Mortezaie et al. To achieve the best results for FR, we also require expertise in the subject of psychology, so that we can study the feature characteristics of the face. Lots of work has been done on the FR from the standard algorithms, like Principal Component Analysis (PCA), Local Binary Pattern (LBP) to the latest DL methods, like Convolutional Neural Networks (CNNs).

The organization of the paper is as follows. In Section 2, we provide the main steps involved in the process of FR. In Section 3, we summarize the various FR algorithms based on ML and DL. In Section 4, we provide open challenges and directions for future scope and in section 5, we conclude the work.

2. STEPS INVOLVED IN THE PROCESS OF FR

FR can be considered as a way of authentication and verification. In this sence, a new unknown face is matched with various other faces present in the database which all have known entities. After this

N. Singhal, V. Ganganwar, M. Yadav, A. Chauhan, M. Jakhar and K. Sharma are with the Department of Computer Engineering, Army Institute of Technology, Pune, India. Emails: [[1]ngupta, [2]vganganwar]@aitpune.edu.in, [[3]yadavmenka2009, [4]ashachauhan8085, [5]jakharmahender8 and [6]karinasharma1119]@gmail.com)]

comparison, a result is given out signifying whether the face has been recognized or not. The face identity is either confirmed or denied by the result drawn after comparison with the face data present in the database. FR process consists of two major components to carry out the whole process that is face detection and FR.

2.1 Face Detection

It involves detecting a face or all faces in a given image or video by using various detection techniques. Its robustness to pose, illumination and the elimination of background results in better detection of faces.

The Viola-Jones [2] is the most used face detector that is based on Haar-like features and shows better results for front faces in its real-time implementation. Some deep learning-based methods are also used for detecting faces, like the sliding-window idea [3], R-CNN [4] and the single-shot detector (SSD) [5] that are successfully used for face detection and provide good results.

2.2 Face Recognition

FR is one of the crucial parts where the detected face after conversion in grayscale is recognized and compared with certain images for authentication or identification. It takes the image detected as input and then checks it with the images present in the database for validation.

The detected face is compared against the database features and if there is a match, then the face is recognized properly and if not, authenticity is not provided to that user.

3. Algorithms Used for FR

3.1 PCA-based

It is a statistical approach, where a set of possibly correlated variables' observations are converted into linearly uncorrelated variables' values known as principal component using orthogonal transformation. Here, data is transformed with help of its projection that is further treated as principal component for first coordinate and then such more variance is created called second component, … and so on resulting in a new coordinate system.

PCA has many advantages when applied to ML algorithms for factors like dimensionality reduction, feature transformation, data visualization as well as for Speeding up the machine learning algorithms and showing better results in terms of recognition rate compared to other techniques, especially to recognize faces with expression disturbance and background disturbance.

PCA and its combination with other algorithms have been widely used for FR application in the past few decades. Wang et al. [6] used LBP and PCA with ABAS algorithm combination for FR. The LBP and PCA were combined used as the feature extraction method and the ABAS algorithm was used for optimization of the neural network, while softmax function was used here to reduce the time for multi-face classification that was constructed to carry out the FR process. Here, the ORL [7] dataset was used to test the proposed model to showcase its capability to handle multi-face classification.

Wang et al. [8] used the F-2D-QPCA technique for FR. They used F-norm to maximize image variance and a greedy iterative algorithm for good convergence and robustness of the method. Experimental results of this model on several color face image databases showed its effectiveness and accuracy compared to other existing models, as their method uses an image as a quaternion matrix that uses color and spatial information of an image.

Kong et al. [9] proposed the CSGF (2D) 2PCANet algorithm for FR. The proposed model used CSGF to overcome the computation time and data redundancy problems of existing models. It consisted of one stage for non-linear output for which linear SVM was used and two stages for feature extraction that had good locality, for which two-dimensional PCA was used. The proposed model showed a higher recognition rate when it was tested on AR [10], ORL databases …etc. and had stable robustness to face image variations' resulting in improving the accuracy of the model.

Low et al. [11] presented a method to boost the performance of 2FGFC against other face descriptors by using the standard 40 multi-scale multi-orientation Gabor filters into the condensed Gabor filter ensemble of only 8 filters. The demodulated Gabor phase features are grasped by an average pooling operator followed by whitening PCA to obtain the final representation with better performance.

Zhang et al. [12] presented a reliable PCA-based FR outsourcing protocol. The proposed methods for privacy-preserving matrix addition, multiplication and vector multiplication ensured the safety of the used technique. The Freivalds algorithm was used here for result verification. The proposed method benefited in rapid development in FR while ensuring the security of the application.

Table 1 summarizes face recognition based on PCA and Figure 1 shows the performance of different PCA algorithms along with ML and DL methods on YALE B and AR databases.

Table 1. PCA-based methods for face recognition.

Study & Pub. Year	Method/Algorithm	Dataset	Accuracy
[6], 2020	Novel Multi-face Recognition, ABASNet	ORL, ExtYaleB [14] and FERET [15]	99.35%, 99.54% and 99.18%, respectively on three datasets
[8], 2020	A Quaternion PCA Method, F-2D-QPCA	GT [16], GT-noise, GT-outlier, FT and FT-outlier	F-2D-QPCA-72.29%, QRR-72.29%, QSR-71.86% (for 30 features)
[9], 2018	Deep Learning, CSGF(2D)^{2}PCANet	XM2VTS [17], ORL, Extend YaleB, LFW[18] and AR	99.58%, 97.50%, 100%, 98.58%, +97.50%, respectively on the five datasets
[13], 2017	Stacking-based CNNs, PCANet+	FERET, LFW and YTF [19]	94.23%

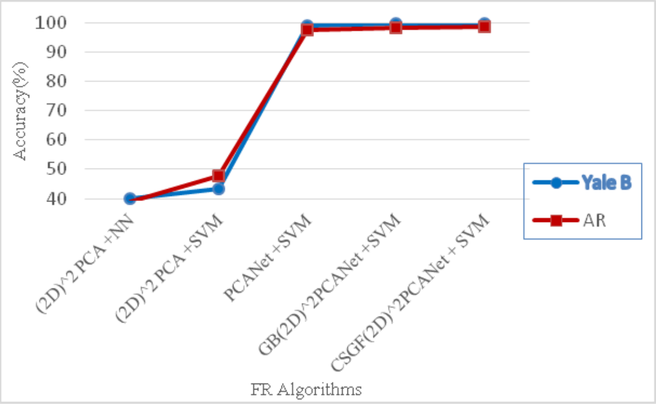


Figure 1. Accuracy of PCA along with ML and DL on YALE B and AR databases.

3.2 LBP-based

It is a texture operator that labels the image's pixels by thresholding each pixel's neighbourhood and using a binary number as a result to represent local features in images. It is considered to be useful for texture classification and improves the detection performance when used with histogram of oriented gradient (HOG).It is robust for monotonic grayscale transformations, gives a great result in a controlled environment and is one of the easiest FR algorithms.

Dalali et al. [20] used discrete wavelet transform as a preprocessing method for extracting significant features. Daubechies wavelets help in extracting approximation coefficients with single-level decomposition, so that the information for FR can be removed. The main focus is to reduce the information into a less significant coefficient, hence resulting in few storage uses. For this paper, the dataset taken into consideration was the MIT face dataset. Two types of performance results were obtained; i.e., for images with noise and for images without noise. For images without noise, an accuracy of 99.3% was achieved and for images with noise, the accuracy was 98.28%.

Tang et al. [21] used the LBP operator to extract features from the face texture and then used 10 CNNs with five different neural structures for more feature extraction for training purposes, as well as for network parameter improvisation and classification results by using the softmax function after the layers were fully connected. Finally, using majority voting, parallel ensemble learning was used to generate the final result of FR. The FR rate in the ORL was increased to 100% using this method, while Yale-B improved it to 97.51 percent.

To reduce the effect of face image variations on feature extraction performance, Muqeet et al. [22] proposed a method that uses directional wavelet transform (DIWT) and LBP to overcome the effect. The LBP histogram features were extracted from selected top-level DIWT sub-bands as a local descriptive feature set. The proposed method was tested on ORL, FEI [23] and GT databases. Results showed that the proposed method was more efficient than LGBP, LSPBPS and CTLBP methods.

Zhang et al. [24] proposed a face anti-spoofing strategy by using LBP, DWT (Discrete Wavelet Transform) and DCT (Discrete Cosine Transform) with an SVM classifier. In this paper, DWT-LBP features were generated that contained information regarding blocks' spatial details of video files and at last, the SVM classifier with RBF kernel was trained for anti-spoofing.

To improve the speed and accuracy of 3-dimensional FR, Shi et al. [25] presented an SVM and LBP combination.LBP was used for feature extraction of details of the 3-D face depth image. After that, information classification was done by SVM. The databases used for experimentation were the Texas 3DFRD 3-D face depth database and self-made depth database. The results showed that the proposed method had a low time consumption and a high recognition rate. Table 2 summarizes face recognition based on LBP.

Table 2. LBP-based methods for face recognition.

Study& P. Year	Method/Algorithm	Dataset	Accuracy
[20], 2016	LBP	MIT face database	Without noise: Max. 99.3%, Min. 99.0% With noise: Max. 98.28%, Min. 97.82%
[21], 2020	CNN and LBP	ORL and Yale-B	ORL: 100%, Yale-B: 97.5%
[22], 2017	LBP	ORL , GT and FEI	ORL: 97%, GT: 82.25%, FEI: 91.14%
[24], 2020	DWT-LBP-DCT	REPLAY-ATTACK [26] and CASIA-FASD [27]	REPLAY-ATTACK: 7.361%, CASIA-FASD: 93.84%
[25], 2020	LBP and SVM	Texas 3DFRD3-D face depth and self-made depth databases.	96.83%

3.3 HOG-based

It is a simple feature descriptor used in image processing to extract features from images and to detect objects. A feature descriptor simplifies an image by extracting only the information that is needed and discarding the rest.HOG features are useful for the first step in detecting objects. Gradient-based representation is obtained from pixel-based representation and is used with linear classification techniques and multi-scale pyramids for object detection.

Zemgulys et al. [28] proposed an image segmentation method using HOG and SVM algorithms for classification. Two approaches were discussed to detect the hand gestures of the referee; i.e., the wearable sensors and computer vision to recognize the signals from the referee in a basketball match, where an accuracy of 97.5% was achieved and the F1-score was 94.95%.

Rameswari et al. [29] implemented an access control system where face detection and recognition were the main parameters for the access control and the HOG was used for feature extraction and facenet algorithm for FR. Along with that, FR RFID technology was used to make the system more secure. In this system, the FaceNet algorithm achieved a higher accuracy of 97% compared other face detection algorithms, like LBPH, FisherFace, …etc.

Chitlangia et al. [30] proposed a method in which the personality trait of an individual is predicted based upon his/her handwriting. HOG was used for feature extraction and those features acted as an input to the SVM model, where classification of the writer's personality traits was done into Introvert, Optimistic, Energetic, Sloppy and Extrovert. The proposed method using the polynomial kernel had 80% accuracy.

Lakshmi et al. [31] used LBP with modified HOG features for facial expression recognition and a multi-class SVM algorithm was used for classification and recognition. Two datasets used were JAFFE [32] and CK+ [33] datasets. The accuracy with CK+ dataset was 97.66%.

Yan et al. [34] used HOG, Adaboost and SVM combinations for the application of real-time vehicle detection. The HOG was used for feature extraction and then the AdaBoost classifier was trained by the combination of HOG features and the dataset that is used in Treatment Group of Images for the classifier training. The accuracy with the HOG and AdaBoost combination was 97.24% and it reached 96.89% using the HOG features with the SVM classifier. Table 3 summarizes face recognition based on HoG.

Table 3. HoG-based methods for face recognition.

Study& P. Year	Method/Algorithm	Dataset	Accuracy
[28], 2018	HOG and SVM	Private	97.5%
[29], 2020	HOG and FaceNet	Private	97%
[30], 2019	HOG and SVM	Private	80%
[31], 2021	Modified HOG along with LBP and SVM	JAFFE and CK+	JAFFE: 90.83% and CK+ : 97.66%
[34], 2016	HOG with AdaBoost and SVM classifier	GTI vehicle [35]	AdaBoost: 97.24% and SVM: 96.89%

3.4 SVM-based

SVM provides a new dimensionality to pattern recognition problems. It can solve face recognition problems with both linear and nonlinear SVM training models. As it requires less computation power, it is commonly used in ML classification problems.

Zhang et al. [36] have extracted multi-scale features from the images of 20 subjects each having different poses with seven different expressions by using bi-orthogonal wavelet entropy to extract multi-scale features. They also employed a strict validation model using stratified cross-validation. They have achieved results superior to three state-of-the-art methods with the accuracy of 96.77% using fuzzy multiclass support vector machines to be classifiers.

The main aspect, according to Pham et al. [37], is to overcome the problem encountered in CNNs when we have imbalanced training data points for classes by increasing the number of training samples of the minority class. They created an image with similar facial expressions using the Action Units (AU) feature set. To improve the model's overall efficiency, AU features are combined with CNN features to train SVM for classification.

Omara et al. [38] developed multimodal biometric systems using hybrid Learning Distance Metric and Directed Acyclic Graph SVM models. The model was tested on an AR face dataset and achieved an accuracy of 99.85%, outperforming many state-of-the-art multi-modal methods. Kernel SVM is used as a classifier which provided better results than traditional classifiers.

Zhang et al. [39] detected athletes' fatigue states by developing an SVM-based model keeping the acceptance criteria of the Sequential Forward Floating Selection (SFFS) algorithm. They used an adaptive median filter method to remove noise and smooth the image and an adaptive threshold light equalization method to adjust the light. The dimensionality of the entire feature set is reduced and a fatigue motion feature subset is extracted. If the face images have more than 80% of their eyes closed, the method classifies them as fatigued. Table 4 summarizes face recognition based on SVM with different techniques.

Table 4. SVM-based methods for face recognition.

Study& P. Year	Method/Algorithm	Dataset	Accuracy
[36], 2016	Fuzzy SVM and Stratified Cross-validation	20 subjects X 7 different expressions (Private)	96.77+-0.10%
[37], 2019	SVM fused with CNN (DenseNet) and AU features	RAF, Fer2013, ExpW	RAF: 91.37%, Fer2013: 71.01%, ExpW: 72.84%
[38], 2021	Distance Metric and DAG SVM	AR face dataset	99.85%
[39], 2020	SVM and SFFS	8000 face images (Private)	Above 90%

3.5 CNN-based

In recent years, CNNs have been recommended to solve computer vision problems as they have shown tremendous growth. The convolution and pooling layers of the CNN can extract the maximum amount of facial features compared to standard algorithms when used for FR. As the amount of training data is increasing in this digital world, we need a deep learning model that takes significantly less amount of time to train the model.

Syafeeza et al. [40] challenged the main factors (illumination variances, poses, facial expressions, occlusions) which affect the performance of the face recognition algorithm by proposing a robust 4-layer CNN architecture. The system achieved an accuracy of 99.5% on AR database and 85.13% on FERET database (on its 35 subjects). The most significant feature of the system was that it takes less than 0.01 second to complete the FR process.

Zangeneh et al. [41] used a coupled mapping method architecture for high- and low-resolution face images that have two branches of deep convolutional neural networks for each type of resolution to be converted into a common space. The branch associated with the conversion to the common space from high-resolution consists of fourteen-layer network, whereas the branch corresponding to low-resolution face image transformation consists of an additional network of 5 layers that was connected to the 14-layer network. It was tested on FERET, LFW and MBGC [42] datasets, where the proposed architecture proved a 5% better accuracy that is 97.2% compared to the traditional methods implemented before and outperformed the other methods, showing good performance when applied to very low-resolution images of 6*6 pixels.

Im et al. [43] proposed an authentication system for preserving the privacy of Smartphone users against malicious clients by storing a feature vector of the face in the encrypted form. Euclidean distance-based matching score is computed whenever someone tries to access the private vector on the remote server. It takes 1.3 seconds to perform the secure face verification in real-time whereas it takes just 1 second for the CFP [44] and ORL datasets to face verification. To further improve the computational score, they used the Catalano-Fiore transformation that converts a linear homomorphic encryption scheme into a quadratic scheme.

Goel et al. [45] have used a high-level method of feature extraction based on the DCNN-Optimized Kernel Extreme Learning Machine algorithm. Particle Swarm Optimization (PSO) algorithm is used for parameter optimization alongside polynomial function Kernel ELM classification algorithm. The results achieved without normalization on the datasets AT&T, CMU-PIE [46], Yale [47] and UMIST [48] were 0.5, 8.89, 0 & 21 error rate. This method has the least training time compared to other DLNs.

Zhao et al. [49] handled various face presentation attacks by proposing a deep architecture to increase the accuracy of multi-view human FR. Here, the authors proposed a CNN for extracting face features and to further localize the key points on the face, it has used the face alignment algorithm. PCA was used for dimensionality reduction of the deep features and a joint Bayesian framework (JBF) was proposed to score the similarity of feature vectors. An accuracy of 98.52% was achieved on CAS-PEAL [50] dataset.

To address the challenge of automatic age estimate in real-time applications, Al-Shannaq et al. [51] proposed a model for estimating human age using a fine-tuned CNN model. Two types of datasets were used to evaluate the idea. The MAE for the FG NET (limited) dataset was 3.446, while the MAE for the UTKFace (unconstrained) dataset was 4.867. Using the Adience dataset, the model was fine-tuned for the age group classification task and the overall accuracy the model achieved was 61.4%. Table 5 summarizes face recognition based on CNN.

Table 5. CNN-based methods for face recognition.

Study& P. year	Method/Algorithm	Dataset used	Accuracy
[40], 2014	4-layer CNN architecture	AR, FERET	AR: 99.5% and FERET: 85.13%
[41], 2019	Coupled mapping method, DCNNs	FERET, LFW and MBGC	FERET: 99.2%, LFW: 76.3% and MBGC: 68.64%
[43], 2020	Euclidean distance-based, Catalano-Fiore transformation	CFP,ORL	EER: 1.17 ,0.37

[45], 2020	OKELM algorithm, PSO, polynomial function KELM	AT&T,CMU-PIE,YALE,UMIST	EER: 0,0,6.67,10.9
[49], 2020	CNN + PCA,JBF	CAS-PEAL	98.52%
[51], 2020	CNN	FG NET, UTKFace, Adience	MAE: 3.446, MAE: 4.867, 61.4%

3.6 AlexNet-based

The name AlexNet refers to a CNN that has a significant impact in the field of DL for computer vision. It comprises of data augmentation, (1111, 55, 33, convolutions), dropout, max pooling, ReLU activations and SGD with momentum.

Suleman Khan et al. [52] proposed an advanced smart-glasses' framework capable of recognizing faces. The use of portable smart glasses to implement facial recognition can assist law-enforcement officials in recognizing a suspect's face. They have an advantage over security cameras due to their portability and superior frontal view capturing. This technique has a detection rate of 98 % when using 3099 features. AlexNet is used for facial recognition and after training 2500 photos in each class, it has achieved 98.5 % accuracy. During recognition, problems such as emotions and light intensity can be overcome by using a large number of different photos.

Hailong Yu et al. [53] proposed a method in which feature extraction is improved by employing an MLP convolutional layer. The CASIA-Web dataset is used for training and testing. After 10575 trials, the model has achieved 82.3% identification rate. For face verification, the LFW face database was used and 6000 pairs of face comparison trials were calculated, yielding an average recognition rate of 84.5%.

Suleman Khan et al. [54] proposed a framework for facial recognition based on AlexNet and transfer learning. This network requires a vast database to train, but the accuracy is great. They used four different classes from the database for training, with 1000 photos in each class and achieved an accuracy of 97.95%. Table 6 summarizes face recognition based on AlexNet.

Table 6. AlexNet-based methods for face recognition.

Study& Publication year	Method/Algorithm	Dataset used	Accuracy
[52], 2019	CNN + AlexNet	2500 variant images in a class using 3099 features	98.5%
[53], 2019	MLP + MFM in CNN	CASIA-Web data LFW face database	82.3% 84.5%
[54], 2019	AlexNet	1000 different people	97.95%

3.7 ResNet-based

A residual neural network (ResNet) is a popular deep learning model that uses residual blocks to overcome the problem of training extremely deep networks. It skips connections and leaps over layers by using these blocks. This skill facilitates the training of huge networks without increasing the percentage of training errors.

Ze Lu et al. [55] proposed a model for low-resolution FR called 'deep coupled ResNet model'. The trunk network, a ResNet-like network, was used to extract the discriminative features shared by face photos of various resolutions. Then, using branch networks, coupled mappings were learned to project features of images. The proposed model was experimented on LFW (with different probe sizes) and SCface datasets (with three different sets of dataset in accordance to camera distance), where it achieved 93.6% - 98.7% and 73.0% - 98.0% accuracy in face verification.

Storey et al. [56] introduced a 3DPalsyNet framework for mouth motion recognition and facial palsy grading in their research. For collecting the dynamic actions of the video data, they used a modified 3D CNN architecture with a ResNet backbone. The structure was tested using two datasets; CK+ and their own Facial Palsy dataset, achieving an F1-score of 82 % for mouth motion and 88% for facial palsy grading, respectively, where these values were greater than those for 3D CNN demonstrating its capacity to perform efficient facial analysis from video sequences.

Peng et al. [57] provided two approaches for FR. The first was to convert Inception-residual ResNet's scaling factor from a hyperparameter to a trainable parameter, with the initial tiny value of 0.1 ensuring a stable training network at the start. The second was to use Leaky ReLU and PReLU in the Inception-ResNet module, which boosts network performance by maximizing input data utilization. Both methods were tested on the VGGFace2, MS1MV2, IJBB and LFW datasets, with better accuracy and training process stability.

Li et al. [58] proposed an enhanced facial emotion recognition model by using ResNet-50 as the network backbone and CNN for feature extraction. To increase the model's convergence ability, BN and the activation function ReLU were used. The model was tested using their own dataset of 20 different subjects (700 images) with varying expressions and ages and it was found to have good accuracy of 95.39 ± 1.41%. Table 7 summarizes face recognition based on ResNet.

Table 7. ResNet-based methods for face recognition.

Study& P. year	Method/Algorithm	Dataset used	Accuracy
[55], 2018	Deep coupled ResNet model	LFW face database & SCface datasets	93.6 - 98.7 % 73.0 - 98.0 %
[56], 2019	Modified 3D CNN + ResNet	CK+ dataset	F1 score: 82%
[57], 2019	Modified 3D CNN + ResNet	Private Facial Palsy dataset	F1 score: 88%
[58], 2021	CNN + ResNet-50	Private dataset of 20 different subjects (700 images)	95.39 ± 1.41%

3.8 Comparison of All Methods

The performances of some FR algorithms which are using commonly used LFW and ORL datasets are shown in Figure 2 and Figure 3. In Figure 2, we compared CSGF(2D)2PCANet [9] with a Linear SVM approach PCANet+ [13] with DL-based AlexNet [53] and ResNet [55] model on LFW as common database, where CSGF(2D)2PCANet was proven to be better with an accuracy rate of 98.58% compared to other models.

Figure 3 shows the comparison of the performances of these algorithms on ORL face dataset. PCANet with a Linear SVM approach [9] yielded 91% accuracy and (2D) PCA with NN in [9] yielded approximately 97% accuracy, CNN with LBP combination [21] gave 100% accuracy, HOG + LDP with linear SVM [59] achieved approximately 97% accuracy and the simple HOG with linear SVM model [59] gave approximately 90% accuracy over this dataset, whereas for CNN, we studied ESPCN with CNN combination [60], which gave approximately 93% accuracy over this dataset. Based on the results, we concluded that the LBP and HoG combination with ML had higher accuracy and performed better than other models on the ORL dataset.

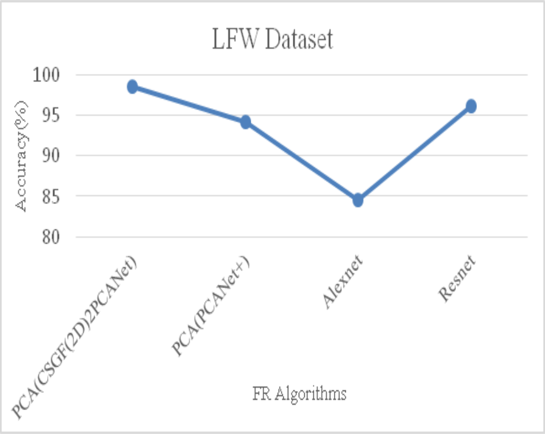


Figure 2. Performance of FR algorithms on LFW database.

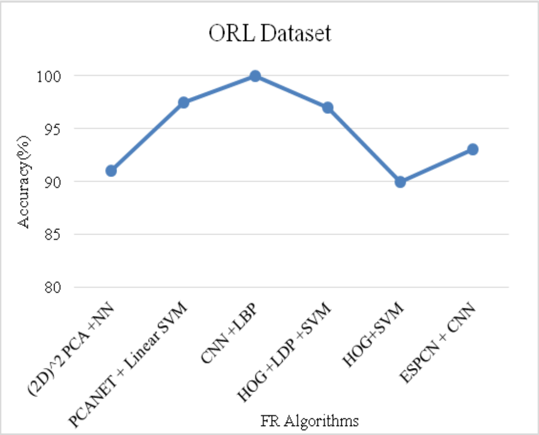


Figure 3. Performance of FR algorithms on ORL database.

3.9 Dataset

Table 8 summarizes the popular datasets used by researchers for FR.

Table 8. Summary of datasets used for FR.

Year	Dataset	Total Images/Videos	Features
1994	ORL [7]	400 images	Various facial expressions, facial details (glasses or no glasses) and lighting conditions were used in the images. With the subjects in a frontal, upright position, a dark, homogenous background was used.
1997	YALE [47]	165 grayscale images	Each subject has 11 photos, one for each different face emotion or configuration: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, drowsy, shocked and wink.
1998	AR [10]	4000 images, 126 persons (56 women and 70 men)	Images consist of different expression (i.e., Neutral expression, Smile, Anger, Scream), various lighting conditions, with and without wearing sunglasses or a scarf.
1998	FERET [15]	14,126 images, 1199 individuals	It consists of 1564 sets of images with 365 duplicate sets of images.
1998	JAFFE [32]	213images	Ten Japanese female models were used to pose for 7 facial expressions (6 fundamental face expressions + 1 neutral).
1998	UMIST [48]	564 images, 20 subjects	Every image features a variety of positions, ranging from profile to frontal views. Subjects represent a diverse spectrum of races and genders resulting in a more comprehensive dataset.
1999	XM2VTS [17]	2360 mug shots, 295 individuals	Dataset is supplied with manually located eye points for all 2360 images for better recognition.
2001	YALE-B [14]	5760 images	Each subject is viewed in 576 different ways (9 poses x 64 illumination conditions).A photograph with ambient (background) illumination was also captured for each individual in a certain stance.
2001	Extend Yale B [14]	2414 images, 38 subjects	Images were taken in a variety of lighting circumstances and with a variety of facial expressions, resulting in an excellent result.
2002	CMU-PIE [46]	41,368 images	Each photograph was taken in 13 distinct stances, with 43 various lighting situations and four distinct expressions.
2003	CAS-PEAL [50]	99594 images	Chinese face database with large-scale images. To gather 27 photos in three shots, each individual is instructed to gaze straight ahead, up and down. The database also includes five facial expressions, six accessories and 15 lighting adjustments.
2007	LFW [18]	13,233 images, 1680 people	Its goal is to collect facial images and other relevant data for Wikipedia's Living People category.
2009	MBGC [42]	628 Videos	There are 4025 frames in which the left iris is visible and 4013 frames in which the right iris is visible. An Iris On the Move (IOM) technology took the near-infrared facial video.
2009	Multi-PIE [61]	750,000 images, 337 people	15 views and 19 lighting settings were used to photograph the subjects with various expressions and frontal images with high resolution.
2010	FEI [23]	2800 images	Faces of people between the ages of 19 and 40, each having a distinct appearance, haircut and adornments were used for images.
2011	YTF [19]	3,425 videos, 1,595 persons	Used YouTube as a source for videos. Each subject has an average of 2.15 videos available. The average length of a video is 181.3 frames with the shortest clip of 48 frames and the longest clip of 6,070 frames.
2012	CASIA-FASD [27]	600 (240 for training and 360 for testing), 50 subjects (12 videos per subject)	Anti-spoofing dataset. Videos were taken under different light conditions and resolutions.
2012	GTI-Vehicle [35]	3425 images of vehicle rears, 3900 images extracted from road sequence	This database has images extracted from a video sequence. The images cover different driving conditions, especially related to weather.

2012	Replay-Attack [26]	1300 videos	It is a face-spoofing database. All videos were created by displaying a snapshot or video recording of the same client for at least 9 seconds or having an actual client try to access a laptop through a built-in webcam.
2016	CFP [44]	500 individuals (each subject has10 frontal and 4 profile images)	It has 10defined splits, each containing 350 same and 350 not-same pairs.

4. Open Challenges for Future Research

The early dataset that was used consisted of images taken under specified and controlled environments. The accuracy of the algorithm is severely affected under adverse conditions of the image, such as low resolution, blur, pose variation and occlusion. Most of the image-based data available today is obtained using low-resolution devices and to get higher accuracy with this data is a challenge. The latest huge datasets created using images from the internet are not annotated properly, which results in poor accuracy of the DNN models. They are also prone to face-spoofing attacks as they can be easily deceived. So, there is a requirement for more robust DNNs. Video-based datasets yield better results as we can capture the dynamic aspects of the face which helps counter spoofing attacks on the networks.

5. Conclusions

In this paper, we explored existing FR techniques based on various descriptor methods combined with machine learning classifiers, such as SVM, deep learning and transfer learning. We also listed the popular datasets used for FR technique. The limitations of existing FR technique are that, they used the datasets of images taken under specified and controlled environments and the performance of these systems degrades under adverse conditions known as semantic adversarial attacks or when downloaded from the internet. Our study will provide insight into existing FR techniques for researchers who wish to conduct their research in this field. The challenge for the future study is to develop a robust FR algorithm that can handle low-resolution images captured in an uncontrolled environment.

References

[1] Z. Mortezaie and H. Hassanpour, "A Survey on Age-invariant Face Recognition Methods," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 05, no. 02, pp. 87-96, August 2019.

[2] W. LU and M. YANG, "Face Detection Based on Viola-Jones Algorithm Applying Composite Features," Proc. of the IEEE Int. Conf. on Robots & Intell. Sys. (ICRIS), pp. 82-85, Haikou, China, 2019.

[3] H. Li, Z. Lin, X. Shen, J. Brandt and G. Hua, "A Convolutional Neural Network Cascade for Face Detection," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 5325-5334, DOI: 10.1109/CVPR.2015.7299170, Boston, USA, 2015.

[4] H. Chen, Y. Chen, X. Tian and R. Jiang, "A Cascade Face Spoofing Detector Based on Face Anti-spoofing R-CNN and Improved Retinex LBP," IEEE Access, vol. 7, pp. 170116-170133, 2019.

[5] X. Hu and B. Huang, "Face Detection Based on SSD and CamShift," Proc. of the IEEE 9th Joint Int. Inform. Techn. and Artificial Intelligence Conf. (ITAIC), pp. 2324-2328, Chongqing, China, 2020.

[6] F. Wang, F. Xie, S. Shen, L. Huang, R. Sun and J. Le Yang, "A Novel Multi Face Recognition Method with Short Training Time and Lightweight Based on ABASNet and H-Softmax," IEEE Access, vol. 8, pp. 175370-175384, DOI: 10.1109/ACCESS.2020.3026421, 2020.

[7] F. S. Samaria and A. C. Harter, "Parameterization of a Stochastic Model for Human Face Identification," Proc. of IEEE Workshop on Applications of Computer Vision, 1994, pp. 138-142, Sarasota, USA, 1994.

[8] M. Wang, L. Song, K. Sun and Z. Jia, "F-2D-QPCA: A Quaternion Principal Component Analysis Method for Color Face Recognition," IEEE Access, vol. 8, pp. 217437-217446, 2020.

[9] J. Kong, M. Chen, M. Jiang, J. Sun and J. Hou, "Face Recognition Based on CSGF(2D)2PCANet," IEEE Access, vol. 6, pp. 45153-45165, DOI: 10.1109/ACCESS.2018.2865425, 2018.

[10] A. Martinez and R. Benavente, "The AR Face Database," CVC Technical Report 24, [Online], Available: http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html, 1998.

[11] C. Low, A. B. Teoh and C. Ng, "Multi-fold Gabor, PCA and ICA Filter Convolution Descriptor for Face Recognition," IEEE Trans. on Circuits and Systems for Video Techn., vol. 29, no. 1, pp. 115-129, 2019.

[12] Y. Zhang, X. Xiao, L. Yang, Y. Xiang and S. Zhong, "Secure and Efficient Outsourcing of PCA-based Face Recognition," IEEE Trans. on Information Forensics and Security, vol. 15, pp. 1683-1695, 2020.

[13] C. Y. Low, A. B. J. Teoh and K. A. Toh, "Stacking PCA Net+: An Overly Simplified Conv Nets Baseline for Face Recognition," IEEE Signal Processing Letters, vol. 24, no. 11, 2017.

[14] A.S. Georghiades, P.N. Belhumeur and D.J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 6, pp. 643–660, 2021.

[15] P. J. Phillips, H. Wechsler, J. Huang and P. J. Rauss, "The FERET Database and Evaluation Procedure for Face-recognition Algorithms," Image and Vision Computing, vol. 16, no. 5, pp. 295-306, 1998.

[16] The Georgia Tech Face Database, [Online], Available: http://www.anefian.com/research/face_reco.htm.

[17] K. Messer, J. Matas, J. Kittler, J. Luettin and G. Maitre, "XM2VTSDB: The Extended M2VTS Database," Proc. of the 2nd International Conference on Audio and Video-based Biometric Person Authentication, vol. 964, pp. 965-966, 2000.

[18] G. B. Huang, M. Mattar, T. Berg and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," Proc. of Workshop on Faces in 'Real-Life' Images: Detection, Alignment and Recognition, [Onlive], Available: https://hal.inria.fr/inria-00321923, Marseille, France, 2008.

[19] L. Wolf, T. Hassner and I. Maoz, "Face Recognition in Unconstrained Videos with Matched Background Similarity," Proc. of CVPR 2011, pp. 529-534, DOI: 10.1109/CVPR.2011.5995566, Colorado Springs, USA, 2011.

[20] S. Dalali and L. Suresh, "Daubechives Wavelet Based Face Recognition Using Modified LBP," Procedia Computer Science, vol. 93, pp. 344-350, 2016, DOI:10.1016/j.procs.2016.07.219, 2016.

[21] J. Tang, Q. Su, B. Su, S. Fong, W. Cao and X. Gong, "Parallel Ensemble Learning of Convolutional Neural Networks and Local Binary Patterns for Face Recognition," Computer Methods and Programs in Biomedicine, vol. 197, DOI: 10.1016/j.cmpb.2020.105622, 2020.

[22] M. A. Muqeet and R. S. Holambe, "Local Binary Patterns Based on Directional Wavelet Transform for Expression and Pose-invariant Face Recognition," Applied Computing and Informatics, vol. 15, no.2, pp. 163-171, DOI: 10.1016/j.aci.2017.11.002, 2019.

[23] C. E. Thomaz and G. A. Giraldi, "A New Ranking Method for Principal Component Analysis and Its Application to Face Image Analysis," Image and Vision Computing, vol. 28, no. 6, pp. 902-913, DOI: 10.1016/j.imavis.2009.11.005, 2010.

[24] W. Zhang and S. Xiang, "Face Anti-spoofing Detection Based on DWT-LBP-DCT Features," Signal Processing: Image Communication, vol. 89, Paper ID: 115990, DOI: 10.1016/j.image.2020.115990, 2020.

[25] L. Shi, X. Wang and Y. Shen, "Research on 3D Face Recognition Method Based on LBP and SVM," Optik, vol. 220, Paper ID: 165157, DOI: 10.1016/j.ijleo.2020.165157, 2020.

[26] I. Chingovska, A. Anjos and S. Marcel, "On the Effectiveness of Local Binary Patterns in Face Anti-Spoofing," Proceedings of the IEEE International Conference of Biometrics Special Interest Group (BIOSIG), pp. 1-7, Darmstadt, Germany, 2012.

[27] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi and S. Z. Li, "A Face Anti Spoofing Database with Diverse Attacks," Proc. of the 5th IAPR IEEE International Conference on Biometrics (ICB), pp. 26-31, DOI: 10.1109/ICB.2012.6199754, New Delhi, India, 2012.

[28] J. Žemgulys, V. Raudonis, R. Maskeliūnas and R. Damaševičius, "Recognition of Basketball Referee Signals from Videos Using Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM)," Procedia Computer Science, vol. 130, pp. 953-960, 2018.

[29] R. Rameswari, S. N. Kumar, M. A. Aananth and C. Deepak, "Automated Access Control System Using Face Recognition," Materials Today: Proceedings, vol. 45, DOI: 10.1016/j.matpr.2020.04.664, 2020.

[30] A. Chitlangia and G. Malathi, "Handwriting Analysis Based on Histogram of Oriented Gradient for Predicting Personality Traits Using SVM," Procedia Computer Science, vol. 165, pp. 384-390, DOI: 10.1016/j.procs.2020.0, 2019.

[31] D. Lakshmi and R. Ponnusamy, "Facial Emotion Recognition Using Modified HOG and LBP Features with Deep Stacked Autoencoders," Microprocessors and Microsystems, vol. 82, DOI: 10.1016/j.micpro.2021.103834, 2021.

[32] M. Lyons, M.Kamachi and J. Gyoba, "TheJapanese Female Facial Expression (JAFFE) Database," The Japanese Female Facial Expression (JAFFE) Dataset, Zenodo, DOI: 10.5281/zenodo.3451524, 1998.

[33] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion-specified Expression," Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 94-101, DOI: 10.1109/CVPRW.2010.5543262, San Francisco, USA, 2010.

[34] G. Yan, M. Yu, Y. Yu and L. Fan, "Real-time Vehicle Detection Using Histograms of Oriented Gradients and AdaBoost Classification," Optik, vol. 127, no. 19, pp. 7941-7951, 2016.

[35] J. Arróspide, L. Salgado and M. Nieto, "Video Analysis Based Vehicle Detection and Tracking Using an MCMC Sampling Framework," EURASIP Journal on Advances in Signal Processing, vol. 2012, Article ID: 2012 (2), DOI: 10.1186/1687-6180-2012-2, 2012.

[36] Y.-D. Zhang, Z.-J. Yang, H.-M. Lu, X.-X. Zhou, P. Philips, Q.-M. Liu and S.-H. Wang, "Facial Emotion Recognition Based on Biorthogonal Wavelet Entropy, Fuzzy Support Vector Machine and Stratified Cross Validation," IEEE Access, vol. 4, pp. 8375-8385, DOI: 10.1109/ACCESS.2016.2628407, 2016.

[37] T. T. D. Pham and C. S. Won, "Facial Action Units for Training Convolutional Neural Networks," IEEE Access, vol. 7, pp. 77816-77824, DOI: 10.1109/ACCESS.2019.2921241, 2019.

[38] I. Omara, A. Hagag, S. Chaib, G. Ma, F. E. Abd El-Samie and E. Song, "A Hybrid Model Combining Learning Distance Metric and DAG Support Vector Machine for Multimodal Biometric Recognition," IEEE Access, vol. 9, pp. 4784-4796, DOI: 10.1109/ACCESS.2020.3035110, 2021.

[39] F. Zhang and F. Wang, "Exercise Fatigue Detection Algorithm Based on Video Image Information Extraction," IEEE Access, vol. 8, pp. 199696-199709, DOI: 10.1109/ACCESS.2020.3023648, 2020.

[40] A. R. Syafeeza, M. Khalil-Hani, S. S. Liew and R. Bakhteri, "Convolutional Neural Network for Face Recognition with Pose and Illumination Variation," International Journal of Engineering and Technology (IJET), vol. 6, no. 1, pp. 44-57, 2014.

[41] E. Zangeneh, M. Rahmati and Y. Mohsenzadeh, "Low Resolution Face Recognition Using a Two-branch Deep Convolutional Neural Network Architecture," Expert Systems with Applications, vol. 139, Article ID: 112854, DOI: 10.1016/j.eswa.2019.112854, 2019.

[42] P. J. Phillips et al., "Overview of the Multiple Biometrics Grand Challenge," Proc. of Advances in Biometrics (ICB 2009), Part of Lecture Notes in Computer Science, vol. 5558. Springer, Berlin, Heidelberg, [Online], Available: https://doi.org/10.1007/978-3-642-01793-3_72, 2021.

[43] J. Im, S. Jeon and M. Lee, "Practical Privacy-Preserving Face Authentication for Smartphones Secure Against Malicious Clients," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2386-2401, DOI: 10.1109/TIFS.2020.2969513, 2020.

[44] S. Sengupta, J. Chen, C. Castillo, V. M. Patel, R. Chellappa and D. W. Jacobs, "Frontal to Profile Face Verification in the Wild," Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1-9, DOI: 10.1109/WACV.2016.7477558, Lake Placid, USA, 2016.

[45] T. Goel and R. Murugan, "Classifier for Face Recognition Based on Deep Convolutional Optimized Kernel Extreme Learning Machine," Computers & Electrical Engineering, vol. 85, Paper ID: 106640, DOI: 10.1016/j.compeleceng.2020.106640, 2020.

[46] T. Sim, S. Baker and M. Bsat, "The CMU Pose, Illumination and Expression Database," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 12, pp. 1615-1618, Dec. 2003.

[47] A. Georghiades, P. Belhumeur and D. Kriegman, "Yale Face Database," Yale University, [Online] Available: http://cvc.yale.edu/projects/yalefaces/yalefa, 1997.

[48] F. Zhao, J. Li, L. Zhang, Z. Li and S. Na, "Multi-view Face Recognition Using Deep Neural Networks," Future Generation Computer Systems, vol. 111, pp. 375-380, DOI: 10.1016/j.future.2020.05.002, 2020.

[49] W. Gao et al., "The CAS-PEAL Large-scale Chinese Face Database and Baseline Evaluations," IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, vol. 38, no. 1, pp. 149-161, DOI: 10.1109/TSMCA.2007.909557, 2008.

[50] A. Al-Shannaq and L. Elrefaei, "Age Estimation Using Specific Domain Transfer Learning," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 02, pp. 122-139, June 2020.

[51] S. Khan, M. H. Javed, E. Ahmed, S. A. A. Shah and S. U. Ali, "Facial Recognition Using Convolutional Neural Networks and Implementation on Smart Glasses," Proc. of the IEEE International Conference on Information Science and Communication Technology (ICISCT), pp. 1-6, Karachi, Pakistan, 2019.

[52] Z. Ren and X. Xue, "Research on Multi Pose Facial Feature Recognition Based on Deep Learning," Proc. of the 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), pp. 1427-1433, DOI: 10.1109/ICMCCE51767.2020.00313, Harbin, China, 2020.

[53] S. Khan, E. Ahmed, M. H. Javed, S. A. A Shah and S. U. Ali, "Transfer Learning of a Neural Network Using Deep Learning to Perform Face Recognition," Proc. of the International Conference on Electrical, Communication and Computer Engineering (ICECCE), pp. 1-5, Swat, Pakistan, 2019.

[54] Z. Lu, X. Jiang and A. Kot, "Deep Coupled ResNet for Low-resolution Face Recognition," IEEE Signal Processing Letters, vol. 25, no. 4, pp. 526-530, April 2018.

[55] G. Storey, R. Jiang, S. Keogh, A. Bouridane and C. Li, "DPalsyNet: A Facial Palsy Grading and Motion Recognition Framework Using Fully 3D Convolutional Neural Networks," IEEE Access, vol. 7, pp. 121655-121664, DOI: 10.1109/ACCESS.2019.2937285, 2019.

[56] S. Peng, H. Huang, W. Chen, L. Zhang, W. Fang, "More Trainable Inception-ResNet for Face Recognition," Neurocomputing, vol. 411, pp. 9-19, DOI: 10.1016/j.neucom.2020.05.022, 2020.

[57] B. Li and D. Lima, "Facial Expression Recognition *via* ResNet-50," International Journal of Cognitive Computing in Engineering, vol. 2, pp. 57-64, DOI: 10.1016/j.ijcce.2021.02.002, 2021.

[58] H. Wang, D. Zhang and Z. Miao, "Fusion of LDB and HOG for Face Recognition," Proc. of the 37th IEEE Chinese Control Conf., pp. 9192-9196, DOI: 10.23919/ChiCC.2018.8483900, Wuhan, China, 2018.

[59] M. A. Talab, S. Awang and S. A. M. Najim, "Super-low Resolution Face Recognition Using Integrated Efficient Sub-pixel Convolutional Neural Network (ESPCN) and Convolutional Neural Network (CNN)," Proc. of the IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), pp. 331-335, DOI: 10.1109/I2CACIS.2019.8825083, Selangor, Malaysia, 2019.

[60] R. Gross, I. Matthews, J. Cohn, T. Kanade and S. Baker, "Multi-PIE," Image and Vision Computing, vol. 28, no.5, pp. 807-813, doi:10.1016/j.imavis.2009.08.002, 2010.

ملخص البحث:

في عالمنا الحاضر، تُستخدم أنظمة القياس الحيويّة لتحليل السِّمات المميّزة الجسديّة او السّلوكيّة لشخصٍ ما من أجل التّصديق أو التّمييز. وحتى الآن، هناك العديد من هذه الانظمة التي تستخدم العيْن أو بصمة الإصبع أو سِمات الوجْه للتّمييز بين الأشخاص والتّحقّق من هويّاتهم؛ إذ تُعتبر الأنظمة القائمة على تمييز الوجوه هي المفضّلة على نطاقٍ واسع، لأنها لا تتطلب مساعدة المُستخدِم في كل وقت، بالإضافة الى أنّها آليّة الى حدٍّ أكبر وسهلة التّشغيل.

هذا البحث عبارة عن ورقة مراجعة تقدّم دراسةً مقارنةً بين تقنيات متنوعة لتمييز الوجوه وتركيباتها الهجينة. كذلك تمّ تحليل أكثر مجموعات البيانات استخداماً في هذا الميدان ومراجعتها، الى جانب تسليط الضّوء على الآفاق المستقبلية والتحدّيات فيما يتعلق بموضوع البحث، والخوارزميات القائمة على التعلُّم العميق المستخدمة في مجال تمييز الوجوه.

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) مجلة علمية عالمية متخصصة محكمة تنشر الأوراق البحثية الأصيلة عالية المستوى في جميع الجوانب والتقنيات المتعلقة بمجالات تكنولوجيا وهندسة الحاسوب والاتصالات وتكنولوجيا المعلومات.

تحتضن جامعة الأميرة سمية للتكنولوجيا (PSUT) المجلة الأردنية للحاسوب وتكنولوجيا المعلومات، وهي تصدر بدعم من صندوق دعم البحث العلمي في الأردن. وللباحثين الحق في قراءة كامل نصوص الأوراق البحثية المنشورة في المجلة وطباعتها وتوزيعها والبحث عنها وتنزيلها وتصويرها والوصول إليها. وتسمح المجلة بالنسخ من الأوراق المنشورة، لكن مع الإشارة إلى المصدر.

الأهداف والمجال

تهدف المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) إلى نشر آخر التطورات في شكل أوراق بحثية أصيلة وبحوث مراجعة في جميع المجالات المتعلقة بالاتصالات وهندسة الحاسوب وتكنولوجيا المعلومات وجعلها متاحة للباحثين في شتى أرجاء العالم. وتركز المجلة على موضوعات تشمل على سبيل المثال لا الحصر: هندسة الحاسوب وشبكات الاتصالات وعلوم الحاسوب ونظم المعلومات وتكنولوجيا المعلومات وتطبيقاتها.

الفهرسة

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات مفهرسة في كل من:

فريق دعم هيئة التحرير

عنوان المجلة

الموقع الإلكتروني: www.jjcit.org
البريد الإلكتروني: jjcit@psut.edu.jo
العنوان: جامعة الاميرة سمية للتكنولوجيا، شارع خليل الساكت، الجبيهة، عمان، الأردن.
صندوق بريد: 1438 عمان 11941 الأردن
هاتف: +962-6-5359949
فاكس: +962-6-7295534

Princess Sumaya University for Technology
Since 1991 تأسست
PSUT
جامعة الأميرة سمية للتكنولوجيا

صندوق دعم البحث العلمي
Scientific Research Support Fund

المجلة الأردنية للحاسوب و تكنولوجيا المعلومات

أيلول ٢٠٢١ المجلد ٧ العدد ٣
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

الصفحات عنوان البحث

٢٢٣ - ٢٣٨ طريقة مبتكرة لنمذجة وتصحيح حقل الانحياز تعتمد على تعدُّد السّطوح في صور الرّنين المغناطيسي
داويا عزوز، و سامين معزوزي

٢٣٩ - ٢٥٢ ابتكار خوارزمية تسيير جديدة للشّبكات اللاسلكية على الدّارات المتكاملة باستخدام تعلّم التّعزيز
زُهرة هاراتي، اسماعيل تاهانيان، على رضا تجاري، و منصور فاتح

٢٥٣ - ٢٦٧ تعزيز الأمان في البرمجيّات العسكريّة ذاتيّة الحماية
كارلوس غونزاليس

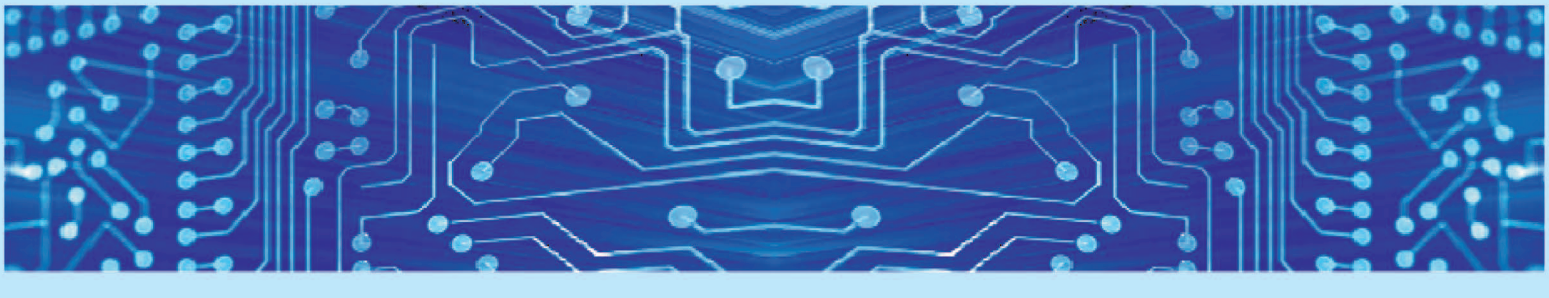
٢٦٨ - ٢٧٧ تقليل اثر تصادُم الوصول العشوائي في الشّبكات اللاسلكية القائمة على إنترنت الأشياء ضيّقة النّطاق
بلال رباح الدوري، و أحمد زُرفي

٢٧٨ - ٢٩١ مرشِّح «كاني» معدَّل لكشف الحواف في الصُّور ذات المدى
محمد شربِت، و سامين معزوزي

٢٩٢ - ٣١٢ إيجاد الحلول المُثلى لمشكلات التّصنيف باستخدام طريقة الذّئاب الرّماديّة الموزونة مع معدّل التقاء محسَّن في تدريب الشّبكات العصبيّة متعدّدة الطبقات
ألوك كومار، لِكراج، و أنوج كومار

٣١٣ - ٣٢٥ دراسة مقارنة لخوارزميات تعلُّم الآلة وخوارزميات التعلُّم العميق لتمييز الوجوه
نكيتا سنغال، فايشالي غانغانوار، منكا ياداف، آشا تشاوهان، باهندر جاكار، و كارينا شارما

jjcit@psut.edu.jo www.jjcit.org



مجلة علمية عالمية متخصصة محكمة

تصدر بدعم من صندوق دعم البحث العلمي