<table>
<thead>
<tr>
<th>PAGES</th>
<th>PAPERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>130 - 149</td>
<td>DISTRIBUTED GREY WOLF OPTIMIZER FOR NUMERICAL OPTIMIZATION PROBLEMS</td>
</tr>
<tr>
<td></td>
<td>Bilal H. Abed-alguni and Malek Barhoush</td>
</tr>
<tr>
<td>150 - 158</td>
<td>ENHANCED ULTRA-WIDE BAND HEXAGONAL PATCH ANTENNA</td>
</tr>
<tr>
<td></td>
<td>Yanal S. Al-Faouri, Noor M. Awad and Mohamed K. Abdelazeez</td>
</tr>
<tr>
<td>159 - 174</td>
<td>ENHANCING THE ACCURACY OF SONBOL’S ARABIC ROOT EXTRACTION ALGORITHM</td>
</tr>
<tr>
<td></td>
<td>Nisrean Thalji, Nik Adilah Hanin, Zyd Thalji and Sohair Al-Hakeem</td>
</tr>
<tr>
<td>175 - 184</td>
<td>FEATURE PRUNING METHOD FOR HIDDEN MARKOV MODEL-BASED ANOMALY DETECTION</td>
</tr>
<tr>
<td></td>
<td>A COMPARISON OF PERFORMANCE</td>
</tr>
<tr>
<td></td>
<td>Sulaiman Alhaidari and Mohamed Zohdy</td>
</tr>
<tr>
<td>185 - 200</td>
<td>SEMI-QUANTITATIVE SECURITY RISK ASSESSMENT OF ROBOTIC SYSTEMS</td>
</tr>
<tr>
<td></td>
<td>Anas AlMajali, Khalil M. Ahmad Yousef, Bassam J. Mohd, Waleed Dweik,</td>
</tr>
<tr>
<td></td>
<td>Salah Abu Ghalyon and Roa’a Hasan</td>
</tr>
</tbody>
</table>

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:

EDITORIAL BOARD SUPPORT TEAM

LANGUAGE EDITOR
Haydar Al-Momani

EDITORIAL BOARD SECRETARY
Eyad Al-Kouz

All articles in this issue are open access articles distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).

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

Ahmad Hiasat (EIC)  Adil Alpkoçak
Adnan Gutub  Adnan Shaout
Arafat Awajan  Christian Boitet
Gheith Abandah  Haytham Bani Salameh
Ismail Ababneh  João Luis Marques Pereira Monteiro
Leonel Sousa  Mohammad Mismar
Omer Rana  Taisir Alghanim

INTERNATIONAL ADVISORY BOARD

Ahmed Yassin Al-Dubai  Albert Y. Zomaya
UK  AUSTRALIA
Chip Hong Chang  Enrique J. Gomez Aguilera
SINGAPORE  SPAIN
Fawaz Al-Karmi  George Ghinea
JORDAN  UK
Gian Carlo Cardarilli  Issam Za’balawi
ITALY  JORDAN
João Barroso  Kareem Sakallah
PORTUGAL  USA
Khaled Assaleh  Laurent-Stephane Didier
UAE  FRANCE
Lewis Mackenzies  Zoubir Hamici
UK  JORDAN
Marc Dacier  Marco Winzker
QATAR  GERMANY
Martin T. Hagan  Marwan M. Krunz
USA  USA
Michael Ullman  Mohammad Alhaj Hasan
USA  JORDAN
Mohammed Benaissa  Mowafaq Al-Omosh
UK  JORDAN
Nadim Obaid  Nazim Madhavji
JORDAN  CANADA
Omar Al-Jarrah  Othman Khalifa
JORDAN  MALAYSIA
Paul G. Plöger  Shahrul Azman Mohd Noah
GERMANY  MALAYSIA
Shambhu J. Upadhyaya  Wejdan Abu Elhaija
USA  JORDAN

“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”.

"ما ورد في هذه المجلة يعتبر عن آراء الباحثين ولا يعكس بالضرورة آراء هيئة التحرير أو الجامعة أو سياسة صندوق دعم البحث العلمي".
DISTRIBUTED GREY WOLF OPTIMIZER FOR NUMERICAL OPTIMIZATION PROBLEMS

Bilal H. Abed-alguni and Malek Barhoush

(Received: 29-Jul.-2018, Revised: 10-Sep.-2018, Accepted: 13-Sep.-2018)

ABSTRACT
The Grey Wolf Optimizer (GWO) algorithm is an interesting swarm-based optimization algorithm for global optimization. It was inspired by the hunting strategy and leadership hierarchy of grey wolves. The GWO algorithm has been successfully tailored to solve various continuous and discrete optimization problems. However, the main drawback of GWO is that it may converge to sub-optimal solutions in early stages of its simulation process due to the loss of diversity in its population. This paper introduces a distributed variation of GWO (DGWO) that attempts to enhance the diversity of GWO by organizing its population into small independent groups (islands) based on a well-known distributed model called the island model. DGWO applies the original GWO to each island and then allows selected solutions to be exchanged among the islands based on the random ring topology and the best-worst migration policy. The island model in DGWO provides a better environment for unfit candidate solutions in each island to evolve into better solutions, which increases the likelihood of finding global optimal solutions. Another interesting feature about DGWO is that it can run in parallel devices, which means that its computational complexity can be reduced compared to the computational complexity of existing variations of GWO. DGWO was evaluated and compared to well-known swarm-based optimization algorithms using 30 CEC 2014 functions. In addition, the sensitivity of DGWO to its parameters was evaluated using 15 standard test functions. The comparative study and the sensitivity analysis for DGWO indicate that it provides competitive performance compared to the other tested algorithms. The source code of DGWO is available at: https://www.dropbox.com/s/2d16i46598a03y0/DistributedGreyWolfOptimizer.zip?dl=0.

KEYWORDS
Grey wolf optimizer, Island-based model, Distributed optimization algorithm, Optimization, Swarm-based optimization.

1. INTRODUCTION
Swarm-based optimization algorithms, such as the Bat algorithm [1]-[2], Cuckoo search [3]-[6], Whale optimization [7]-[9], Butterfly optimization [10]-[14], Grasshopper optimization [15], flower pollination [16] and Particle swarm optimization [1], [17], have been successfully used to solve difficult optimization problems in various fields (e.g., image processing [18]-[19], fuzzy logic [20]-[21], control engineering [22]-[25], scheduling problems [26]-[27]). The Grey Wolf Optimizer (GWO) is an interesting swarm-based optimization algorithm that was recently proposed to solve difficult optimization problems based on the hunting strategy and leadership hierarchy of grey wolves [28].

The GWO simulates the leadership hierarchy of grey wolves based on four hierarchical leaderships: alpha wolf, beta wolf, delta wolf and omega wolf. In addition, GWO simulates the hunting strategy of grey wolves based on three sequential steps: searching for prey, encircling prey and attacking prey. GWO has lately attracted much attention from the optimization community due to its attractive advantages. First, GWO is an efficient optimization algorithm that has a simple structure (Figure 1). Second, the simulation process of GWO is controlled by one key parameter (Section 2.1). Finally, GWO has been successfully tailored to solve various continuous optimization problems as well as discrete optimization problems (Section 2.3).

Like most of the swarm-based optimization algorithms, GWO may converge faster than expected to sub-optimal solutions [29]-[30]. This is because the evolutionary operators of GWO may not adequately preserve the diversity of the population over the course of the simulation process of GWO. Swarm-based optimization algorithms can be in general parallelized to run in different machines.
Interestingly, the parallel swarm-based approach offers a possible solution to the problem of premature convergence of most of the swarm-based optimization algorithms [31]-[32]. This might be because the parallel approach allows the population of candidate solutions of a given optimization problem to be divided into several groups, which provides a better environment for unfit candidate solutions to evolve in each group.

The island model, which is a structured population model, can be integrated with the framework of a swarm-based optimization algorithm to facilitate its parallelization [21], [33]. Using the island model, the population of a parallel swarm-based optimization algorithm can be divided into \( n \) groups (islands), where a swarm-based optimization algorithm is applied independently to the population of each island. These islands periodically exchange selected candidate solutions among each other (i.e., migration process) in an attempt to maintain the diversity of population on each island.

The current paper introduces a Distributed Grey Wolf Optimizer (DGWO) that attempts to enhance the diversity of GWO by organizing its population into small islands based on the island model. DGWO applies the original GWO to the population of each island and then allows selected solutions to be exchanged among the islands based on the random ring topology and the best-worst migration policy.

The rest of the paper is organized as follows: Section 2 provides background information about the Grey Wolf Optimizer algorithm, the distributed island model and related work to the Grey Wolf Optimizer. Section 3 presents and discusses the Distributed Grey Wolf Optimizer algorithm. Section 4 presents the simulation results of the proposed algorithm and finally Section 5 presents the conclusions and future work.

2. PRELIMINARIES

This section briefly summarizes some of the underlying concepts of the grey wolf optimizer (Section 2.1) algorithm and the island model (Section 2.2). This section also provides an overview of recently proposed variations of grey wolf optimizer (Section 2.3).

2.1 Grey Wolf Optimizer

The Grey Wolf Optimizer (GWO) algorithm, which was developed by Mirjalili et al. [28], is an interesting nature-inspired optimization algorithm. GWO uses a simulation model based on the hierarchical leadership and hunting strategy of grey wolves. In the wild, grey wolves are top predators, which means that they are at the top of their food chain, with no natural predators. A pack of wolves is normally composed of 6 to 7 wolves, but it might have up to 15 wolves. In a wolf pack, normally an alpha (\( \alpha \)) male and an alpha female wolves control the pack. The direct followers to the wolves are the beta (\( \beta \)) and delta (\( \delta \)) wolves. The \( \beta \) and \( \delta \) wolves help the \( \alpha \) wolves to control and dominate the other wolves in the pack (omega (\( \omega \)) wolves). The hunting strategy of grey wolves is a three-stage cooperative strategy (tracking and chasing prey, pursuing and encircling prey and attacking prey) [34].

Figure 1 shows the flow of the GWO algorithm. The first step of GWO is to generate a population of wolves (candidate solutions) \( \hat{X}_i (i = 1, 2, ..., N) \) for a given optimization problem. Each candidate solution is composed of \( M \) decision variables \( \hat{X}_i = \{x_1, x_2, ..., x_M\} \). The fitness value of each candidate solution is calculated using the fitness function of the optimization problem to determine the hierarchical structure of the population. The solution with the best calculated fitness value is called the alpha solution (\( \hat{X}_\alpha \)), while the solutions with the second and third best fitness values are respectively called the beta solution (\( \hat{X}_\beta \)) and delta solution (\( \hat{X}_\delta \)). The rest of the solutions in the population are called the omega solutions (\( \hat{X}_\omega \)) [20].

The improvement loop of GWO is composed of three stages: tracking and chasing prey, pursuing and encircling prey and attacking prey. Encircling prey is mathematically modelled in GWO as follows:

\[
\vec{D} = |\vec{C} \cdot \hat{X}_p (t) - \hat{X} (t) |
\]

\[
\hat{X} (t + 1) = \hat{X}_p (t) - \vec{A} \cdot \vec{D}
\]

where \( t \) is the current iteration number, \( \vec{A} \) and \( \vec{C} \) are two coefficient vectors, \( \hat{X}_p \) is the candidate solution
that represents the prey and $\hat{X}(t)$ is the candidate solution that represents the grey wolf.

The two coefficient vectors $\hat{A}$ and $\hat{C}$ can be updated as follows:

$$\hat{A} = 2\hat{a} \cdot \hat{r}_1 - \hat{a}$$  \hspace{1cm} (3)$$

$$\hat{C} = 2 \cdot \hat{r}_2$$ \hspace{1cm} (4)

where $\hat{a}$ is a vector with values that linearly decrease from 2 to 0 over the course of the simulation of GWO and $\hat{r}_1$ and $\hat{r}_2$ are two random vectors between 0 to 1.

In GWO, the $\omega$ solutions are updated based on the $\alpha$, $\beta$ and $\delta$ solutions. Equations 5 to 11 represent the update process of the solutions:

$$\hat{D}_{\alpha} = |\hat{C}_{1, \hat{X}_{\alpha}} - \hat{X}|$$ \hspace{1cm} (5)$$

$$\hat{D}_{\beta} = |\hat{C}_{2, \hat{X}_{\beta}} - \hat{X}|$$ \hspace{1cm} (6)$$

$$\hat{D}_{\delta} = |\hat{C}_{3, \hat{X}_{\delta}} - \hat{X}|$$ \hspace{1cm} (7)$$

$$\hat{X}_{1} = \hat{X}_{\alpha} - \hat{A}_{1} \cdot (\hat{D}_{\alpha})$$ \hspace{1cm} (8)$$

$$\hat{X}_{2} = \hat{X}_{\beta} - \hat{A}_{2} \cdot (\hat{D}_{\beta})$$ \hspace{1cm} (9)$$

$$\hat{X}_{3} = \hat{X}_{\delta} - \hat{A}_{3} \cdot (\hat{D}_{\delta})$$ \hspace{1cm} (10)$$

$$\hat{X}(t + 1) = \frac{\hat{X}_{1} + \hat{X}_{2} + \hat{X}_{3}}{3}$$ \hspace{1cm} (11)

Approaching and attacking prey (exploitation stage) is simulated in GWO by decreasing the components of $\hat{a}$ from 2 to 0 over the course of simulation of GWO. According to Equation 3, the vector $\hat{a}$ controls the range of values of $\hat{A}$ which are in the range $[-2\hat{a}, 2\hat{a}]$. It is worth noting that a candidate solution is updated in the direction of the best solutions ($\alpha$, $\beta$ and $\delta$ solutions) when $|\hat{A}| < 1$. The exploration of the search space is triggered in GWO using two settings. First, the candidate solutions diverge from the best solutions when $|\hat{A}| > 1$. Second, the components of $\hat{C}$, which are in the range $[0, 2]$, provide weights for the best solutions in order to increase the influence of the best solutions $|\hat{C}| < 1$ or decrease their influence $|\hat{C}| > 1$ in Equation 1.

\begin{center}
\begin{verbatim}
Initialize the population of $n$ candidate solutions $\hat{X}_i (i = 1, 2, ..., n)$
Initialize $\hat{A}$, $\hat{a}$ and $\hat{C}$
Calculate the fitness value of each candidate solution
$\hat{X}_{\alpha} =$ the best candidate solution
$\hat{X}_{\beta} =$ the second best candidate solution
$\hat{X}_{\delta} =$ the third best candidate solution
While ($t < \text{Max number of iterations}$)
    for each candidate solution
        Update the values of the current candidate solution using Equation 11
    end for
Update $\hat{A}$, $\hat{a}$ and $\hat{C}$
Calculate the fitness value of each candidate solution
Update $\hat{X}_{\alpha}$, $\hat{X}_{\beta}$ and $\hat{X}_{\delta}$
$t = t + 1$
end while
return $\hat{X}_{\alpha}$
\end{verbatim}
\end{center}

Figure 1. The Grey Wolf Optimizer (GWO) Algorithm.

It is important to note that GWO in its current form can be only applied directly to continuous optimization problems. GWO has been tailored for many real-world discrete optimization problems, such as feature selection [35], economic load dispatch problems [36]-[37] and scheduling problems [38].
The computational complexity of GWO (Figure 1) can be calculated as follows:

1. Initializing the population of $n$ candidate solutions requires $n$ operations.
2. Initializing the parameters of GWO ($\tilde{A}$, $\tilde{a}$ and $\tilde{C}$) requires 3 operations.
3. Calculating the first three best solutions ($\tilde{X}_{\alpha}$, $\tilde{X}_{\beta}$, and $\tilde{X}_{\delta}$) requires $n$ operations.
4. The following operations are conducted inside the while loop:
   a. The for loop that is used to update the population requires $n$ operations.
   b. Calculating the fitness of each candidate solution in an island requires $n$ operations.
   c. Updating $\tilde{X}_{\alpha}$, $\tilde{X}_{\beta}$ and $\tilde{X}_{\delta}$ requires $n$ operations.
   d. Updating the parameters of GWO ($\tilde{A}$, $\tilde{a}$ and $\tilde{C}$) requires 3 operations.

Overall, the while loop requires $m(n + n+ 3 + l)$ operations, where $m$ is the maximum number of iterations. The number of operations can be further simplified to $m \cdot n$.

5. All the operations of the algorithm can be calculated as $n + 3+n + m \cdot n$, which can be simplified to $m \cdot n$, because $2n$ is greater than 3 and $m \cdot n$ is greater than $2n$.

In summary, the computational complexity of GWO is $O(m \cdot n)$. Note that any basic vector operation has been assumed to cost $O(1)$ in the above analysis.

### 2.2 Island Model

The island model, which is a distributed population model, can be integrated with the framework of a swarm-based optimization algorithm to facilitate its parallelization [33]. Using the island model, the population of a swarm-based optimization algorithm can be divided into $n$ groups (islands). Each island is assigned to a computation device, where a swarm-based optimization algorithm is applied to its population. An interaction process, called migration, is periodically triggered among the islands in the island model. In the migration process, selected candidate solutions are exchanged among islands in an attempt to maintain and amend the diversity of population on each island. The best-worst and random-random policies are the most used migration policies in the literature [31]-[33]. In the best-worst policy, the most fitted solutions in one island (say $m$ solutions) are exchanged with the $m$ worst fitted solutions in a neighbouring island. In the random-random policy, $m$ random solutions in one island are swapped with $m$ random solutions in a neighbouring island.

The islands in the island model are normally organized and arranged based on a given migration topology [40]. Star, random-star, mesh, random-mesh, ring and random-ring topologies are some popular migration topologies used with swarm-based algorithms [41]. The prefix "random" in the name of the migration topology, which indicates that the order of the islands in the topology changes each time the migration process is triggered.

Two parameters control the migration process among islands in the island model [42]. First, the migration frequency ($M_f$), which determines the number of iterations between two consecutive migration waves. Second, the migration rate ($M_r$), which is a parameter that determines the percentage of solutions to be exchanged between an island and a neighbouring island.

### 2.3 Variations of Grey Wolf Optimizer

Several hybrid GWO algorithms have been recently proposed in an attempt to control and amend premature convergence of GWO. Jayabarathi et al. [36] proposed a hybrid GWO algorithm that integrates the genetic operators (crossover and mutation) of the genetic algorithm (GA) into GWO to improve its exploration ability. The experimental results in [36] suggested that the hybrid GWO and GA algorithm provides good performance in solving several instances of the economic dispatch problem. However, the proposed hybrid algorithm requires heavy computations compared to the basic GWO. This is expected, because applying the genetic operators at each iteration of GWO for each candidate solution is a time-consuming process. Another disadvantage of the hybrid GWO algorithm is that it has a complex structure compared to the GWO algorithm. Tawhid and Ali [29] integrated the whole GA algorithm into GWO to solve the minimization problem of the energy function of the molecule. Although the hybrid GA and GWO algorithm performs much better than the basic GWO, it requires more computational time than GWO. Jitkongchuen [43] proposed a hybrid swarm-based algorithm,
between the differential evolution (DE) algorithm and GWO, for solving numerical optimization functions. The simulation results in [43] suggested that the hybrid GWO is more reliable and more accurate in solving difficult numerical optimization problems than DE, self-adaptive DE and particle swarm optimization (PSO). Unfortunately, the hybrid GWO and DE algorithm is complex and requires heavy computations compared to GWO. Zawbaa et al. [44] combined Antlion optimization (ALO) and GWO in one algorithm (ALO-GWO). ALO-GWO was particularly designed to solve the feature selection problem in large datasets. The simulation results in [44] indicated that ALO-GWO provides good performance in solving feature extraction problems in large datasets compared to PSO and GA. However, ALO-GWO may require a long processing time as most of the hybrid optimization algorithms.

Various intelligent techniques have been successfully used with GWO to enhance its convergence behaviour. Saremi et al. [45] suggested the EPD-GWO algorithm that uses the evolutionary population dynamics (EPD) technique to improve the diversity of population in GWO. In other words, EPD is used in EPD-GWO to improve the exploration of candidate solutions. EPD repositions the worst candidate solutions in the population into the neighbourhoods of the alpha, beta, delta and omega wolves. The simulation results in [45] indicated that EPD-GWO provides better results than GWO. However, the computational complexity of EPD-GWO is higher than the computational complexity of GWO because EPD has to be repeated at each iteration of EPD-GWO. Rodríguez et al. [46] proposed a dynamic variation of GWO that dynamically tunes the parameters of GWO during the simulation process of GWO in order to obtain the best possible performance out of GWO. However, manipulating the parameters of GWO as in [46] does not provide significant enhancement in the performance than the original GWO algorithm. The enhanced GWO (EGWO) algorithm, that was proposed by Joshi and Arora [47], is another adaptive variation of GWO. EGWO dynamically changes the values of the key parameter of GWO (α) over the course of its simulation. Moreover, EGWO enhances the exploitation mechanism of GWO by making the best use of the best solution (alpha solution). The experimental results of EGWO compared to standard optimization algorithms (PSO, Firefly Algorithm (FA) and Flower Pollination Algorithm (FPA)) proves that EGWO is a competitive algorithm for solving constrained optimization problems. Malik et al. [48] introduced the wdgWO algorithm which combines the weighted distance (wd) technique with GWO. In wdgWO, the update strategy of the candidate solutions in GWO is modified and the average function of best solutions is replaced with the weighted sum of best solutions. According to the experimental results in [48], the wdgWO showed superior performance compared to basic optimization algorithms (FA, Artificial bee colony, Cuckoo search and PSO).

Moreover, Emary et al. [49] introduced the experienced Grey Wolf Optimizer (EGWO) algorithm that incorporates reinforcement learning (RL) and artificial neural networks (ANNs) into GWO to improve its performance. EGWO uses RL to update the parameters of each candidate solution in the population of GWO. EGWO uses ANNs to estimate the experience of each candidate solution. The expertise estimation of a solution is used to control its exploration rate. EGWO was evaluated and compared to three optimization algorithms (GWO, PSO, GA). Joshi and Arora [50] proposed an enhanced GWO (E-GWO) algorithm that uses an improved hunting mechanism to balance between the exploration and exploitation of candidate solutions in GWO. Kohli and Arora [51] proposed the chaotic GWO (CGWO) algorithm that incorporates the chaos theory into GWO in an attempt to improve the convergence behaviour of GWO for constrained optimization problems. In CGWO, several types of chaotic maps are employed to adjust the main parameter of GWO (α). The simulation results of CGWO compared to well-known algorithms (PSO, Firefly Algorithm (FA) and Flower Pollination Algorithm (FPA)) suggest that CGWO provides good results compared to the other algorithms. Heidari and Pahlavani [52] introduced a modified GWO that uses the greedy selection method and the Lévy flight operator with GWO in an attempt to improve the exploration mechanism of GWO. The simulation results in [52] indicated that the improved GWO is more reliable and more effective in solving both discrete and continuous optimization problems than popular state-of-the-art swarm-based optimization algorithms. However, the greedy selection method is not the best selection mechanism according to Abed-alguni and Alkhatteeub [4]. Moreover, using the greedy selection method in an optimization algorithm may cause premature convergence [4], [53]. Gupta and Deep [54]-[55] presented an improved GWO algorithm called RW-GWO that uses random walk to enhance the search ability of grey wolves. According to the simulation results in [54]-[55], RW-GWO shows high efficiency in solving both continuous and discrete optimization algorithms.
In summary, the computational complexity of most of the hybrid GWO algorithms (e.g. hybrid GA and GWO [29, 36], DE-GWO [43], EGWO [49], ALO-GWO [44]) is much higher than the computational complexity of the original GWO. This is because the integrated search method in GWO is normally repeated at each iteration of GWO. Moreover, hybrid GWO algorithms have complex structures compared to the basic GWO algorithm. Other enhanced versions of the GWO algorithm (e.g. EPDGWO [45], GWO and Lévy flight operator [52], Dynamic GWO [46], wdGWO [48], EGWO [47]) provide insignificant enhancement in the performance compared to GWO or can be applied only to specific applications. In the next Section, a distributed variation of GWO (DGWO) is introduced in an attempt to enhance the diversity of GWO by organizing its population into small islands based on the island model. An interesting feature about DGWO is that it can run in parallel devices, which means that its computational complexity can be reduced compared to the computational complexity of hybrid GWO algorithms. However, DGWO can be directly applied to continuous optimization problems. DGWO should be tailored to be applicable to real-world discrete optimization problems.

3. DISTRIBUTED GREY WOLF OPTIMIZER

The Grey Wolf Optimizer (GWO) is a nature-inspired optimization algorithm that mimics the hunting strategy and leadership hierarchy of grey wolves [28]. GWO has been applied successfully to various continuous optimization problems [45], [48], [52] and discrete optimization problems [29], [36], [48], [52]. A problem with GWO is that its improvement loop may not maintain the diversity of its population due to the imperfection of its evolutionary operators. Such a problem is a common problem with all optimization algorithms.

The island model enhances the performance and run-time of optimization algorithms. It also provides better chances for unfit solutions in each island to evolve and improve. The distributed genetic algorithm [56], distributed differential evolution [57]-[58], distributed particle swarm optimization [59] and distributed ant colony [60] algorithms are but few examples of successful island-based optimization algorithms.

The current section introduces a Distributed Grey Wolf Optimizer (DGWO) algorithm in an attempt to improve the diversity of GWO by organizing its population into small islands based on the island model. DGWO applies the original GWO to the population of each island and then allows selected solutions to be exchanged among the islands based on the random-ring topology and the best-worst migration policy.

Figure 2 shows the pseudo code of the DGWO algorithm. The first step of DGWO is to determine the total number of candidate solutions \((n)\) for a given number of islands \((s)\) and the maximum number of iterations \((MaxIter)\) of GWO. The second step is to initialize the parameters of the island model. The next step is to generate \(k\) candidate solutions for each island and then to calculate the number of migration waves \((M_n)\) and number of migrant solutions \((n_i)\).

DGWO generates \(k\) candidate solutions for each island before the beginning of the evolution process of DGWO. In DGWO, the evolution process of the basic GWO algorithm is synchronously applied to each individual island. After each \(M_i\) iterations (migration frequency), a number of candidate solutions are swapped between each two neighbouring islands based on the random-ring topology and the best-worst migration policy. In the random-ring topology, the neighbouring relationships are unidirectional relationships (Figure 3). However, the neighbouring relationships in the random-ring topology change after each migration wave among the islands. The number of candidate solutions to be exchanged among the islands is specified by the migration rate \((M_i)\).

The migration process among islands takes place each time the maximum number of iterations that is specified by \(M_i\) is reached. Before the beginning of the migration process, the islands in DGWO are organized to form a unidirectional ring based on the principles of the random ring topology. The most fitted solutions in one island (say \(m\) solutions) are exchanged with the \(m\) worst fitted solutions in a neighbouring island based on the best-worst migration policy. Let \(ppoi = \{x_{i1}^j, x_{i2}^j, \ldots, x_{ik}^j\}\) and \(ppoj = \{x_{j1}^i, x_{j2}^i, \ldots, x_{jk}^i\}\) be the neighboring islands \(I_i\) and \(I_j\), respectively. If we assume that \(R_m = 20\%\), \(n = 150\) and \(s = 5\), the number of migrant solutions is \(R_m \times (n/s) = 20\% \times (150/5) = 6\). Let \(ppoi\) and \(ppoj\) be two lists ordered in ascending order based on their objective values, where \(f(x_{i1}^j) \leq f(x_{i2}^j) \leq \cdots \leq f(x_{ik}^j)\) and \(f(x_{j1}^i) \leq f(x_{j2}^i) \leq \cdots \leq f(x_{jk}^i)\). If we assume that there is a unidirectional edge from \(I_i\) to \(I_j\), the first
six candidate solutions in \( I_i (x_i^1, x_i^2, x_i^3, x_i^4, x_i^5, x_i^6) \) will be exchanged with the last six solutions in \( I_j \{x_j^{s-5}, x_j^{s-4}, x_j^{s-3}, x_j^{s-2}, x_j^{s-1}, x_j^s\} \).

Determine the total number of candidate solutions of all islands \((n)\) and the maximum number of iterations \((\text{MaxIt})\).

Initialize the parameters of the island model \((\text{number of islands } (s), \text{migration frequency } (M_f), \text{migration rate } (M_r))\)

Calculate the population size for each island \((k=n/s)\)

Calculate the number of migration waves \((M_w=\text{MaxIt}/M_f)\)

Calculate the number of migrant solutions \((n_r=n/s \times M_r)\)

Initialize the population of \(k\) candidate solutions for each island \(X_i^j (i = 1,2,...,k)\) and \((j = 1,2,...,s)\)

\(\text{for } i = 1 \text{ to } M_w \) \(.................................(1)\)

\(\text{for } j = 1 \text{ to } s \) \(.................................(2)\)

\(\text{Initialize } \vec{A}_j, \vec{a}_j \text{ and } \vec{C}_j\)

\(t=0\)

Calculate the fitness value of each candidate solution

\(\tilde{X}_\alpha^j = \text{the best candidate solution in island } j\)

\(\tilde{X}_\beta^j = \text{the second best candidate solution in island } j\)

\(\tilde{X}_\delta^j = \text{the third best candidate solution in island } j\)

\(\text{While } (t<M_f)\) \(.................................(3)\)

\(\text{for each candidate solution in island } j \) \(.................................(4)\)

\(\text{Update the values of the current candidate solution using Equation 11} \)

\(\text{end for}\)

Update \(\vec{A}_j, \vec{a}_j \text{ and } \vec{C}_j\)

Calculate the fitness value of each candidate solution in island \( j \)

Update \(\tilde{X}_\alpha^j, \tilde{X}_\beta^j \text{ and } \tilde{X}_\delta^j\)

Replace the \(n_r\) best candidate solutions in island \( j \) with the \(n_r\) worst candidate solutions in island \(((j+1) \mod s)\)

\(t=t+1\)

\(\text{end while}\)

\(\text{end for}\)

\(\text{end for}\)

Calculate \(\tilde{X}_\alpha^j (\tilde{X}_\alpha^j \text{ with the best fitness})\)

\(\text{return the } \tilde{X}_\alpha^j\)

Figure 2. The Distributed Grey Wolf Optimizer (DGWO) Algorithm.

Figure 3. Random-ring Migration Topology.
The computational complexity of DGWO (Figure 2) can be calculated based on its main steps as follows:

1. The first five steps require 5 operations.
2. Initializing the population of \( n \) candidate solutions for all islands requires \( n \) operations.
3. Calculating the fitness of all candidate solutions requires \( n \) operations.
4. Calculating the first three best solutions in an island requires \( k \) operations.
5. The most inner loop (number 4) that is used to update the population of an island requires \( k \) operations.
6. Calculating the fitness of each candidate solution in an island requires \( k \) operations.
7. Updating the first three best solutions in an island requires \( k \) operations.
8. Replacing the \( n \) best candidate solutions in an island with the \( n \) worst candidate solutions in another island requires \( k + k + n \) operations, which can be simplified to \( k \) because \( k > n \).
9. Overall, the while loop (number 3) requires \( M_\ell (k + k + n) \) operations. This step can be simplified to \( M_\ell k \), because \( k > n \).
10. For loop number 2 requires \( s (k + k + M_\ell k) \) operations, which can be simplified to \( s M_\ell k \), because \( s M_\ell k \) is larger than \( s k \).
11. For loop number 1 requires \( M_w s M_\ell k \) operations.
12. The last step that calculates the best candidate solution in all islands requires \( s \) operations.
13. All the operations of the algorithm can be calculated as \( 5 + 2n + k + M_w s M_\ell k + s \), which can be simplified to \( M_w s M_\ell k \) operations.

In summary, the computational complexity of DGWO is \( O(M_w s M_\ell k) \). On the other hand, the computational complexity of the basic GWO is \( O(m n) \), where \( m \) is the maximum number of iterations and \( n \) is the number of candidate solutions. Thus, it is better to run DGWO in parallel devices which will reduce the computational complexity of DGWO to \( O((M_w s M_\ell k) / s) \), where \( s \) is the number of parallel devices (number of islands). In this case, the complexity of DGWO can be simplified to \( O(M_w M_\ell k) \). Note that any basic vector operation has been assumed to cost \( O(1) \) in the above analysis.

### 4. Experiments

In this section, the DGWO is benchmarked on 15 test functions (Table 1 and Table 2). These functions are standard test functions used widely by the researchers to evaluate and compare the performance of swarm-based evolutionary algorithms [3]-[4], [28], [43], [48], [61]. Section 4.1 shows the experimental setup. Section 4.2 provides an analysis of the performance of DGWO based on different experimental scenarios. Section 4.3 provides analysis about the performance of DGWO compared to recently proposed optimization algorithms (Grey Wolf Optimizer (GWO) [28], Cuckoo search (CS) [3], adaptive differential evolution with linear population size reduction evolution (L-SHADE) [62], memory-based hybrid Dragonfly (MHDA) [63] and Fireworks algorithm with differential mutation (FWA-DM) [64].

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Function name</th>
<th>Range</th>
<th>D</th>
<th>( f(\hat{x}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>Generalized Schwefel's Problem 2.26</td>
<td>([-500,500])</td>
<td>30</td>
<td>(-18,983 \times D )</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>Griewank's Function</td>
<td>([-10,10])</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>Whitley's Function</td>
<td>([-10,10])</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>Ackley's Function 2.9</td>
<td>([32.768, 32.768])</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>( f_5 )</td>
<td>Alpine's Function</td>
<td>([-10,10])</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>( f_6 )</td>
<td>Schaffer's Function</td>
<td>([-100,100])</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>( f_7 )</td>
<td>Rastrigin's Function 2.5</td>
<td>([-5.12, 5.12])</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>( f_8 )</td>
<td>Inverted Cosine Wave Function</td>
<td>([-5, 5])</td>
<td>30</td>
<td>(-D+1 )</td>
</tr>
<tr>
<td>( f_9 )</td>
<td>Levy Function</td>
<td>([-10, 10])</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>( f_{10} )</td>
<td>Schwefel's 2.22 Function</td>
<td>([-100,100])</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>( f_{11} )</td>
<td>Rotated Hyper-ellipsoid Function</td>
<td>([-65,536, 65,536])</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>( f_{12} )</td>
<td>Shifted Sphere Function</td>
<td>([-100,100])</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>( f_{13} )</td>
<td>Shifted Schwefel Function</td>
<td>([-100,100])</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>( f_{14} )</td>
<td>Shifted Rastrigin’s Function</td>
<td>([-5, 5])</td>
<td>30</td>
<td>(-330 )</td>
</tr>
<tr>
<td>( f_{15} )</td>
<td>Shifted Expanded Griewank’s Plus</td>
<td>([-5, 5])</td>
<td>30</td>
<td>(-130 )</td>
</tr>
</tbody>
</table>
Table 2. Mathematical formulae of test functions.

<table>
<thead>
<tr>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1(X) = - \sum_{i=1}^{n} [x_i \sin(\sqrt{</td>
<td>x_i^2</td>
</tr>
<tr>
<td>( f_2(X) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 )</td>
<td></td>
</tr>
<tr>
<td>( f_3(x_1 \cdots x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left( \frac{(100(x_i^2 - x_j)^2 + (1-x_j)^2)^2}{4000} - \cos(100(x_i^2 - x_j)^2 + (1-x_j)^2) + 1 \right) )</td>
<td></td>
</tr>
<tr>
<td>( f_4(X) = -20 e^{-\left(0.2 \times \sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}\right)} - e^{\left(0.1 \sum_{i=1}^{n} \cos(2\pi x_i)\right)} + 20 + e^{(1)} )</td>
<td></td>
</tr>
<tr>
<td>( f_5(X) = \sum_{i=1}^{n}</td>
<td>x_i \sin(x_i)</td>
</tr>
<tr>
<td>( f_6(x, y) = 0.5 + \frac{\sin^2(x^2 + y^2) - 0.5}{(1 + 0.001(x^2 + y^2))^2} )</td>
<td></td>
</tr>
<tr>
<td>( f_7(X) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10] )</td>
<td></td>
</tr>
<tr>
<td>( f_8(x) = -\sum_{i=1}^{n-1} -\left(\frac{[x_i^2]_{i+1} + 0.5x_i [x_i+1]}{u}\right) \cos(4 \times \sqrt{x_i^2 + x_i+1 + 0.5x_i [x_i+1]}) )</td>
<td></td>
</tr>
<tr>
<td>( f_9(x) = \sin^2(\pi w_i) + \sum_{i=1}^{d-1} (w_i - 1)^2[1 + 10\sin^2(\pi w_i + 1)] + (w_d - 1)^2[1 + \sin^2(2\pi w_d)] )</td>
<td>where ( w_i = 1 + \frac{x_i-1}{4} ), for all ( i = 1, 2, ..., d )</td>
</tr>
<tr>
<td>( f_{10}(X) = \sum_{i=1}^{n}</td>
<td>x_i</td>
</tr>
<tr>
<td>( f_{11}(X) = \sum_{i=1}^{n} \left( \sum_{j=1}^{i} x_j \right)^2 )</td>
<td></td>
</tr>
<tr>
<td>( f_{12}(X) = \sum_{i=1}^{n} x_i^2 + f_{\text{bias}} = \pi )</td>
<td>where ( z = x - o ) and ( f_{\text{bias}} = 450 )</td>
</tr>
<tr>
<td>( f_{13}(x) = \sum_{i=1}^{d} \left( \sum_{j=1}^{i} x_j^2 \right)^2 + f_{\text{bias}} )</td>
<td>where ( z = x - o ) and ( f_{\text{bias}} = 450 )</td>
</tr>
<tr>
<td>( f_{14}(x) = \sum_{i=1}^{d} (x_i^2 - 10\cos(2\pi x_i) + 10) + f_{\text{bias}} )</td>
<td>where ( z = x - o ) and ( f_{\text{bias}} = 330 )</td>
</tr>
</tbody>
</table>

The definition of \( f_{15} \) is given in [66], where \( z = x - o \) \( f_{\text{bias}} = 130 \)

4.1 Experimental Setup

In each algorithm, the maximum number of iterations was 10,000 and the size of population was 30. For the GWO and DGWO algorithms, the values of vector \( \vec{a} \) were linearly decreased from 2 to 0 over the course of their simulation process. The parameters of the island model in DGWO were tested for different values as shown in Table 3. The parameters of each test function are shown in Tables 1 and 2. The percentage of abandonment in CS was 25% as suggested in [4, [66]. The parameters of L-SHADE (external archive size, historical memory size, control parameter) were dynamically tuned as in [62]. The parameter setting of MHDA was taken from reference [63]. The parameters of FWA-DM were set as follows: \( A_{\text{init}} \) (initial amplitude)= 0.2, \( A_{\text{final}} \) (final amplitude)= 0.001, \( F \) (scaling factor)= 0.5 and \( CR \) (Crossover Operator)= 0.9 as in [64].
4.2 Analysis of the Parameters of DGWO

Table 3 shows nine experimental scenarios used to investigate the relationships between the performance of DGWO and the parameters of the island model ($s$, $M_f$, $M_r$). The purpose of the first three scenarios is to investigate the relationship between the island number $s$ and the performance of DGWO. Scenarios 3-6 aim to investigate the influence of the migration frequency $M_f$ on the performance of DGWO. The purpose of the last three scenarios is to study the influence of the migration rate $M_r$ on the performance of DGWO.

Table 3. Scenarios for analyzing the parameters of DGWO.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$s$</th>
<th>$M_f$</th>
<th>$M_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>2</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>5</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>10</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>Scenario 4</td>
<td>10</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>Scenario 5</td>
<td>10</td>
<td>100</td>
<td>20</td>
</tr>
<tr>
<td>Scenario 6</td>
<td>10</td>
<td>500</td>
<td>20</td>
</tr>
<tr>
<td>Scenario 7</td>
<td>10</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>Scenario 8</td>
<td>10</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>Scenario 9</td>
<td>10</td>
<td>50</td>
<td>30</td>
</tr>
</tbody>
</table>

Tables 4-6 show the simulation results of the nine experimental scenarios illustrated in Table 3. The best results in the tables (lowest objective values) are highlighted with bold. The results were averaged over 50 independent runs.

Table 4. Simulation results of the algorithms for 15 test functions, $D=30$, runs=50, iterations=10,000 (Scenario 1, Scenario 2, Scenario 3).

<table>
<thead>
<tr>
<th>Function</th>
<th>Scenario 1 ($s=2$)</th>
<th>Scenario 2 ($s=5$)</th>
<th>Scenario 3 ($s=10$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>-1.56E+06</td>
<td>-2.08E+06</td>
<td>-2.10E+06</td>
</tr>
<tr>
<td>$f_2$</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>$f_3$</td>
<td>4.57E+01</td>
<td>4.52E+01</td>
<td>4.49E+01</td>
</tr>
<tr>
<td>$f_4$</td>
<td>1.98E+01</td>
<td>4.44E-16</td>
<td>4.44E-16</td>
</tr>
<tr>
<td>$f_5$</td>
<td>6.44E-220</td>
<td>1.97E-222</td>
<td>1.64E-210</td>
</tr>
<tr>
<td>$f_6$</td>
<td>5.00E-01</td>
<td>5.00E-01</td>
<td>5.00E-01</td>
</tr>
<tr>
<td>$f_7$</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>$f_8$</td>
<td>-9.00E+00</td>
<td>-9.00E+00</td>
<td>-9.00E+00</td>
</tr>
<tr>
<td>$f_9$</td>
<td>1.32E+00</td>
<td>1.32E+00</td>
<td>8.52E-01</td>
</tr>
<tr>
<td>$f_{10}$</td>
<td>7.27E+03</td>
<td>1.15E+04</td>
<td>4.21E+03</td>
</tr>
<tr>
<td>$f_{11}$</td>
<td>8.17E+07</td>
<td>5.27E+07</td>
<td>1.36E+07</td>
</tr>
<tr>
<td>$f_{12}$</td>
<td>1.00E+08</td>
<td>1.00E+08</td>
<td>1.00E+08</td>
</tr>
<tr>
<td>$f_{13}$</td>
<td>1.00E+08</td>
<td>1.00E+08</td>
<td>1.00E+08</td>
</tr>
<tr>
<td>$f_{14}$</td>
<td>-2.33E+02</td>
<td>-2.38E+02</td>
<td>-2.40E+02</td>
</tr>
<tr>
<td>$f_{15}$</td>
<td>2.49E+04</td>
<td>3.52E+03</td>
<td>3.15E+03</td>
</tr>
</tbody>
</table>

Table 4 shows the simulation results of DGWO under different island sizes ($s=2$, $s=5$, $s=10$). The results in the table show that DGWO in scenario 3 ($s=10$) performed better than in scenario 1 ($s=2$) and scenario 2 ($s=5$). The results clearly indicate that the performance of DGWO improves with an increase in the number of islands. This is expected, because the existence of many islands (large value of $s$) allows different search regions to be explored simultaneously, while the existence of few islands (low value of $s$) means that few number of search regions can be explored simultaneously [31]-[32]. In addition, the population in a small island would most likely converge to suboptimal solutions earlier than expected [42]. It is worth pointing out that choosing a large value of $s$ increases the computational complexity of DGWO. For example, if the number of islands is 10, then GWO will be repeated 10 times at each iteration of DGWO. Thus, it is better to run DGWO in parallel devices to reduce its computational complexity. Based on the results in Table 4, $s=10$ was used in Tables 5 and 6.
The convergence curves of the first three scenarios of DGWO for three functions \((f_1, f_{11}, f_{15})\) are shown in Figure 4. It is obvious that convergence increases with the increase in iterations for all functions. Figure 4(a), Figure 4(b) and Figure 4(c) show that DGWO in scenario 3 is the fastest converging algorithm.

![Convergence Plots](image)

Figure 4. Convergence Plots of DGWO for Scenarios 1-3.

Table 5 shows the simulation results of DGWO under different migration frequencies \((M_f = 50, M_f = 100, M_f = 500)\). It is obvious that DGWO in scenario 4 \((M_f = 50)\) performed the best followed by DGWO in scenario 2 \((M_f = 100)\) and then DGWO in scenario 3 \((M_f = 500)\). These results suggest that low migration frequencies improve the performance of DGWO compared to high migration frequencies. Basically, low migration frequencies provide more chances for a reasonable number of candidate solutions from the source island to move to the destination island with limited effect on the candidate solutions in the destination island. Consequently, the likelihood that convergence will take longer time to occur increases [31]-[32]. Based on the illustrated results in Table 5, \(M_f = 50\) was used in Table 6.
Table 5. Simulation results of the algorithms for 15 test functions, D=30, runs=50, iterations=10,000 (Scenario 4, Scenario 5, Scenario 6).

<table>
<thead>
<tr>
<th>Function</th>
<th>Scenario 4 ($M_f = 50$)</th>
<th>Scenario 5 ($M_f = 100$)</th>
<th>Scenario 6 ($M_f = 50$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>-2.27E+06</td>
<td>-2.10E+06</td>
<td>-2.13E+06</td>
</tr>
<tr>
<td>$f_2$</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>$f_3$</td>
<td>4.49E+01</td>
<td>4.49E+01</td>
<td>4.47E+01</td>
</tr>
<tr>
<td>$f_4$</td>
<td>4.44E-16</td>
<td>4.44E-16</td>
<td>4.44E-16</td>
</tr>
<tr>
<td>$f_5$</td>
<td>2.38E-224</td>
<td>1.64E-221</td>
<td>2.74E-218</td>
</tr>
<tr>
<td>$f_6$</td>
<td>5.00E-01</td>
<td>5.00E-01</td>
<td>5.00E-01</td>
</tr>
<tr>
<td>$f_7$</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>$f_8$</td>
<td>-9.00E+00</td>
<td>-9.00E+00</td>
<td>-9.00E+00</td>
</tr>
<tr>
<td>$f_9$</td>
<td>7.26E+01</td>
<td>8.52E-01</td>
<td>1.57E+03</td>
</tr>
<tr>
<td>$f_{10}$</td>
<td>4.90E+03</td>
<td>4.21E+03</td>
<td>9.74E+03</td>
</tr>
<tr>
<td>$f_{11}$</td>
<td>1.40E+07</td>
<td>1.36E+07</td>
<td>7.76E+07</td>
</tr>
<tr>
<td>$f_{12}$</td>
<td>9.49E+06</td>
<td>1.00E+08</td>
<td>1.00E+08</td>
</tr>
<tr>
<td>$f_{13}$</td>
<td>1.00E+08</td>
<td>1.00E+08</td>
<td>1.00E+08</td>
</tr>
<tr>
<td>$f_{14}$</td>
<td>-2.60E+02</td>
<td>-2.40E+02</td>
<td>-2.44E+02</td>
</tr>
<tr>
<td>$f_{15}$</td>
<td>1.10E+03</td>
<td>3.15E+03</td>
<td>9.01E+03</td>
</tr>
</tbody>
</table>

Figure 5 shows the convergence curves of the second three scenarios of DGWO for three functions (Figure 5(a) ($f_1$), Figure 5(b) ($f_{11}$), Figure 5(c) ($f_{15}$)). Figure 5(a) and Figure 5(c) show that DGWO in scenario 4 converges faster than the other algorithms, while Figure 5(b) shows that DGWO in scenario 5 converges faster to a solution compared to the other algorithms.
Figure 5. Convergence Plots of DGWO for Scenarios 4-6.

Table 6 shows the simulation results of DGWO under different migration rates ($M_r = 10\%$, $M_r = 20\%$, $M_r = 30\%$). It can be clearly observed that DGWO in scenario 8 ($M_r = 20\%$) performed better than DGWO in scenario 7 ($M_r = 10\%$) and DGWO in scenario 9 ($M_r = 30\%$). These results suggest that there is no clear indication whether high values or low values of $M_r$ improve the performance of DGWO. A possible explanation for the results is that replacing a large number of candidate solutions in an island has an unclear effect on the diversity of its population, while replacing few candidate solutions in an island can act as a seed to enhance its diversity but with a limited effect [31]-[32].

Table 6. Simulation results of the algorithms for 15 test functions, $D=30$, runs=50, iterations=10,000 (Scenario 7, Scenario 8, Scenario 9).

<table>
<thead>
<tr>
<th>Function</th>
<th>Scenario 7 $M_r = 10%$</th>
<th>Scenario 8 $M_r = 20%$</th>
<th>Scenario 9 $M_r = 30%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>-2.75E+06</td>
<td>-2.27E+06</td>
<td>-2.19E+06</td>
</tr>
<tr>
<td>$f_2$</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>$f_3$</td>
<td>4.26E+01</td>
<td>4.49E+01</td>
<td>4.12E+01</td>
</tr>
<tr>
<td>$f_4$</td>
<td>4.44E-16</td>
<td>4.44E-16</td>
<td>4.44E-16</td>
</tr>
<tr>
<td>$f_5$</td>
<td>2.24E-219</td>
<td>2.38E-224</td>
<td>1.83E-223</td>
</tr>
<tr>
<td>$f_6$</td>
<td>5.00E-01</td>
<td>5.00E-01</td>
<td>5.00E-01</td>
</tr>
<tr>
<td>$f_7$</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>$f_8$</td>
<td>-9.0E+00</td>
<td>-9.0E+00</td>
<td>-9.0E+00</td>
</tr>
<tr>
<td>$f_9$</td>
<td>8.52E-01</td>
<td>7.26E-01</td>
<td>1.32E+00</td>
</tr>
<tr>
<td>$f_{10}$</td>
<td>1.72E+04</td>
<td>4.90E+03</td>
<td>1.65E+04</td>
</tr>
<tr>
<td>$f_{11}$</td>
<td>2.91E+07</td>
<td>1.40E+07</td>
<td>3.15E+06</td>
</tr>
<tr>
<td>$f_{12}$</td>
<td>1.00E+08</td>
<td>9.49E+06</td>
<td>4.52E+07</td>
</tr>
<tr>
<td>$f_{13}$</td>
<td>1.00E+08</td>
<td>1.00E+08</td>
<td>1.00E+08</td>
</tr>
<tr>
<td>$f_{14}$</td>
<td>-2.54E+02</td>
<td>-2.6E+02</td>
<td>-2.46E+02</td>
</tr>
<tr>
<td>$f_{15}$</td>
<td>1.00E+03</td>
<td>1.10E+03</td>
<td>1.14E+03</td>
</tr>
</tbody>
</table>

Figure 6 shows the convergence curves of the last three scenarios of DGWO for three functions (Figure 6(a) ($f_1$), Figure 6(b) ($f_{11}$), Figure 6(c) ($f_{13}$)). Figure 6(a) and Figure 6(c) show that DGWO in scenario 7 converges faster than the other algorithms, while Figure 6(b) shows that DGWO in scenario 9 converges faster to a solution compared to the other algorithms. Note that DGWO in Scenario 8 achieved the second best convergence speed for the three functions.

In conclusion, the overall experimental results in this section indicate that high values of $s$ and low values of $M_r$ improve the performance of DGWO. However, there is no clear indication whether high values or low values of $M_r$ improve the performance of DGWO.

4.3 Comparison among DGWO and Other Algorithms

The single-objective real-parameter optimization-benchmark suit of CEC2014 is composed of 30 functions (Table 7). This suit represents an approximation of real-world optimization problems. The search range of each function in the suit is [-100,100]. More details about these functions are available in [67].

Using the single-objective real-parameter optimization-benchmark suit of CEC2014 with 30 dimensions (30 decision variables) [67], the performance of DGWO (Scenario 8) was compared with the performance of GWO and recently proposed optimization algorithms: Cuckoo search (CS), adaptive differential evolution with linear population size reduction evolution (L-SHADE), memory-based hybrid Dragonfly algorithm (MHDA) and Fireworks algorithm with differential mutation (FWA-DM).
Table 7. Single-objective real-parameter optimization-benchmark suit of CEC2014.

<table>
<thead>
<tr>
<th>Function</th>
<th>Function Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$, $f_3$</td>
<td>Unimodal functions</td>
</tr>
<tr>
<td>$f_4$, $f_6$</td>
<td>Multimodal functions</td>
</tr>
<tr>
<td>$f_{17}$, $f_{22}$</td>
<td>Hybrid functions</td>
</tr>
<tr>
<td>$f_{23}$, $f_{30}$</td>
<td>Composite functions</td>
</tr>
</tbody>
</table>

Table 8 shows the function error value (FEV) for the 30CEC2014 functions. The FEVs is the distance between the average of best objective values found in all runs and the true optimal value. Note that the lowest FEV for each function (best result) is marked with Bold. The simulation results in Table 8 show that L-SHADE outperforms the other optimization algorithms by providing the lowest FEV for 11 functions of the 30 functions. This is expected, because L-SHADE is a dynamic differential evolution algorithm that dynamically adjusts its internal parameters and population size over the course of its iterations. Interestingly, DGWO is the second best performing optimization algorithm by producing the lowest FEV for 10 functions of the 30 test functions. This is because DGWO synchronously applies GWO to multiple islands, which accelerates its convergence to a good solution.

Table 8. Simulation results of DGWO compared to five optimization algorithms. D= 30, runs=50, number of iterations is 10,000.

<table>
<thead>
<tr>
<th>Function</th>
<th>GWO</th>
<th>DGWO (Scenario 8)</th>
<th>CS</th>
<th>L-SHADE</th>
<th>MHDA</th>
<th>FWA-DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>2.00E+03</td>
<td>4.36E+00</td>
<td>3.47E+07</td>
<td>9.00E-15</td>
<td>3.59E+03</td>
<td>4.91E+05</td>
</tr>
<tr>
<td>$f_2$</td>
<td>2.12E+03</td>
<td>2.36E+00</td>
<td>2.50E+07</td>
<td>8.50E-11</td>
<td>3.82E+03</td>
<td>2.50E-16</td>
</tr>
<tr>
<td>$f_3$</td>
<td>2.89E-01</td>
<td>2.54E-04</td>
<td>4.10E+04</td>
<td>5.83E-10</td>
<td>5.80E-07</td>
<td>1.88E-16</td>
</tr>
<tr>
<td>$f_4$</td>
<td>8.75E-03</td>
<td>1.63E-09</td>
<td>4.22E+02</td>
<td>2.58E-09</td>
<td>1.42E-08</td>
<td>2.23E+01</td>
</tr>
<tr>
<td>$f_5$</td>
<td>3.96E+02</td>
<td>2.00E+02</td>
<td>5.00E+01</td>
<td>2.00E+01</td>
<td>2.36E+00</td>
<td>2.11E+01</td>
</tr>
<tr>
<td>$f_6$</td>
<td>5.19E+01</td>
<td>1.21E+00</td>
<td>3.63E+01</td>
<td>1.25E-06</td>
<td>8.52E-14</td>
<td>1.82E+01</td>
</tr>
<tr>
<td>$f_7$</td>
<td>2.89E-03</td>
<td>8.53E-10</td>
<td>1.86E+00</td>
<td>7.25E-09</td>
<td>2.25E-11</td>
<td>2.53E-03</td>
</tr>
<tr>
<td>$f_8$</td>
<td>1.33E+00</td>
<td>1.51E-19</td>
<td>3.89E+02</td>
<td>1.25E-09</td>
<td>2.20E-19</td>
<td>9.53E-15</td>
</tr>
<tr>
<td>$f_9$</td>
<td>1.82E+01</td>
<td>1.03E+00</td>
<td>3.00E+03</td>
<td>8.96E+00</td>
<td>5.30E+00</td>
<td>6.54E+01</td>
</tr>
<tr>
<td>$f_{10}$</td>
<td>9.79E+00</td>
<td>3.20E-03</td>
<td>4.37E+03</td>
<td>2.36E-02</td>
<td>1.22E+03</td>
<td>1.13E+01</td>
</tr>
<tr>
<td>$f_{11}$</td>
<td>1.99E+04</td>
<td>2.95E+03</td>
<td>4.00E+03</td>
<td>2.30E+03</td>
<td>1.52E+02</td>
<td>2.19E+03</td>
</tr>
<tr>
<td>$f_{12}$</td>
<td>8.50E+00</td>
<td>6.30E-02</td>
<td>4.78E-01</td>
<td>9.00E-01</td>
<td>1.42E-01</td>
<td>3.25E-01</td>
</tr>
<tr>
<td>$f_{13}$</td>
<td>2.19E+00</td>
<td>4.59E-01</td>
<td>4.81E-01</td>
<td>6.50E-01</td>
<td>4.78E-01</td>
<td>3.11E-01</td>
</tr>
<tr>
<td>$f_{14}$</td>
<td>2.35E-01</td>
<td>1.99E-01</td>
<td>4.28E-01</td>
<td>8.60E-01</td>
<td>5.43E-01</td>
<td>2.99E-01</td>
</tr>
<tr>
<td>$f_{15}$</td>
<td>1.01E+02</td>
<td>7.23E+01</td>
<td>9.94E+01</td>
<td>1.60E+00</td>
<td>3.25E+00</td>
<td>8.36E+00</td>
</tr>
<tr>
<td>$f_{16}$</td>
<td>1.90E+01</td>
<td>9.53E+00</td>
<td>1.53E+01</td>
<td>1.02E+01</td>
<td>1.06E+01</td>
<td>1.10E+01</td>
</tr>
<tr>
<td>$f_{17}$</td>
<td>1.66E+02</td>
<td>4.55E+03</td>
<td>3.47E+06</td>
<td>2.20E+00</td>
<td>4.53E+02</td>
<td>6.59E+03</td>
</tr>
<tr>
<td>$f_{18}$</td>
<td>8.77E+00</td>
<td>3.94E+01</td>
<td>3.90E+03</td>
<td>1.90E+00</td>
<td>3.69E+00</td>
<td>7.24E+01</td>
</tr>
<tr>
<td>$f_{19}$</td>
<td>4.96E+01</td>
<td>1.22E+02</td>
<td>6.14E+01</td>
<td>5.30E+00</td>
<td>3.78E+02</td>
<td>1.04E+01</td>
</tr>
<tr>
<td>$f_{20}$</td>
<td>6.20E+01</td>
<td>4.73E+02</td>
<td>3.97E+04</td>
<td>4.30E+00</td>
<td>7.09E+02</td>
<td>4.37E+01</td>
</tr>
<tr>
<td>$f_{21}$</td>
<td>1.04E+03</td>
<td>7.09E+02</td>
<td>3.57E+05</td>
<td>3.69E+02</td>
<td>2.57E+02</td>
<td>8.75E+02</td>
</tr>
<tr>
<td>$f_{22}$</td>
<td>2.42E+02</td>
<td>2.73E+02</td>
<td>9.47E+02</td>
<td>1.32E+02</td>
<td>2.73E+02</td>
<td>1.62E+02</td>
</tr>
<tr>
<td>$f_{23}$</td>
<td>3.65E+02</td>
<td>3.69E+01</td>
<td>3.78E+02</td>
<td>3.26E+02</td>
<td>3.10E+03</td>
<td>3.16E+02</td>
</tr>
<tr>
<td>$f_{24}$</td>
<td>2.24E+02</td>
<td>2.25E+02</td>
<td>2.89E+02</td>
<td>1.93E+02</td>
<td>2.26E+02</td>
<td>2.96E+02</td>
</tr>
<tr>
<td>$f_{25}$</td>
<td>2.45E+02</td>
<td>2.11E+02</td>
<td>3.26E+02</td>
<td>2.00E+02</td>
<td>2.11E+02</td>
<td>2.09E+02</td>
</tr>
<tr>
<td>$f_{26}$</td>
<td>3.29E+02</td>
<td>2.10E+02</td>
<td>2.22E+02</td>
<td>2.69E+02</td>
<td>1.00E+02</td>
<td>9.93E+01</td>
</tr>
<tr>
<td>$f_{27}$</td>
<td>2.95E+02</td>
<td>4.09E+02</td>
<td>5.22E+02</td>
<td>1.26E+02</td>
<td>4.05E+02</td>
<td>4.10E+02</td>
</tr>
<tr>
<td>$f_{28}$</td>
<td>5.36E+02</td>
<td>1.65E+03</td>
<td>3.86E+03</td>
<td>3.62E+02</td>
<td>1.54E+03</td>
<td>4.22E+02</td>
</tr>
<tr>
<td>$f_{29}$</td>
<td>2.39E+02</td>
<td>2.29E+02</td>
<td>2.59E+05</td>
<td>7.33E+02</td>
<td>7.86E+02</td>
<td>2.78E+02</td>
</tr>
<tr>
<td>$f_{30}$</td>
<td>3.32E+02</td>
<td>2.83E+00</td>
<td>2.39E+04</td>
<td>6.99E+02</td>
<td>2.63E+03</td>
<td>4.69E+02</td>
</tr>
</tbody>
</table>
This indicates that DGWO performs well compared to powerful optimization algorithms. Note that GWO and CS have the worst performance compared to the other algorithms. A possible explanation is that CS and GWO do not employ any special convergence-enhancement technique compared to the other tested algorithms.

5. CONCLUSIONS

The current paper presented the Distributed Grey Wolf Optimizer (DGWO) algorithm that is based on a distribution model called the island model. The population in DGWO is divided into small populations in an attempt to enhance the diversity of the population and the run-time of the algorithm. DGWO applies the original GWO to the population of each island and then allows selected solutions to be exchanged among the islands based on the random ring topology and the best-worst migration policy.

Different experimental cases were designed and used to study the sensitivity of the performance of DGWO to the parameters of the island model (number of islands $s$, migration frequency $M_f$ and migration rate $M_r$). The overall experimental results suggest that high values of $s$ and low values of $M_f$ significantly improve the performance of DGWO. However, there is no clear indication whether high values or low values of $M_r$ improve the performance of DGWO.

In addition, 30 functions of CEC2014 (real-parameter single-objective optimization-benchmark suit) have been used to compare the performance of DGWO to the performance of well-known optimization algorithms (CS, L-SHADE, MHDA, FWA-DM). The results indicate that DGWO produces competitive results compared to those produced by the other compared algorithms. Interestingly, DGWO produced the lowest FEV for 10 functions of the 30 test functions of CEC2014. This is expected, because DGWO synchronously applies GWO to several islands, which accelerates its convergence to good solutions. Moreover, DGWO provides better chances for unfit candidate solutions in each island to evolve to better candidate solutions.

There are four interesting directions for future work. First, it would be interesting to incorporate the island model to multi-objective discrete GWO [38] to explore its efficiency in solving the scheduling problem in welding production. This scheduling problem is one of the most time-consuming processes in modern industry. Second, a binary version of DGWO will be developed and used to solve the problem of feature selection [35], [68]. Feature selection is normally considered as a complex time-consuming problem when it is used with large datasets. Third, hierarchical Q-learning [69]-[70] and cooperative Q-learning [71]-[72] require heavy and complex computations to efficiently solve learning problems with large state space or action space. Based on the fact that the population of Q-values (i.e., values of state-action pairs in Q-learning) in Q-learning can be represented as an optimization problem [1], [17], [66] and [71], the DGWO algorithm will be applied to hierarchical Q-learning [69]-[70] and cooperative Q-learning [71]-[72] as discussed in [17], [66]. Finally, the experimental results in Section 4.3 demonstrated that L-SHADE [62] performs better than many powerful optimization algorithms. Therefore, the incorporation of the island model into the L-SHADE algorithm will be addressed in a future study in an attempt to elevate the performance of L-SHADE.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Oguz Emrah Turgut for the source code of GWO that he made publically, [Online], available at: https://www.researchgate.net/publication/281939423_Grey_Wolf_Optimizer_java_code.

REFERENCES


J. Liang, B. Qu and P. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-parameter Numerical Optimization," Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China; and Technical Report, Nanyang Technological University, Singapore, 2013.


تم استخدام "الخوارزمية "الذكاء الاصطناعي" من الخوارزميات المدمجة المستخدمة في الأمثلة والمfinity على تقنية السويت. وقد استوحيت من استراتيجيات الطائر والقرود التي تتيح للذكاء الاصطناعي مزود قوة في حل المشكلات المعقدة. وتم استخدام تجارب لحل مشاكل محددة من المشكلات المستمرة والمجردة للحل العملية الأمثلة. ومع ذلك، فإن السلبية الرئيسية لهذا تمكن في أنه قد تجمع فيHighlighted Text in Document

未曾提到的英文文献


ملخص البحث:

تعود تعدد "الخوارزمية "الذكاء الاصطناعي" من الخوارزميات المدمجة المستخدمة في الأمثلة والمfinity على تقنية السويت. وقد استوحيت من استراتيجيات الطائر والقرود التي تتيح للذكاء الاصطناعي مزود قوة في حل المشكلات المعقدة. وتم استخدام تجارب لحل مشاكل محددة من المشكلات المستمرة والمجردة للحل العملية الأمثلة. ومع ذلك، فإن السلبية الرئيسية لهذا تمكن في أنه قد تجمع فيHighlighted Text in Document

未曾提到的英文文献


ملخص البحث:

تعود "الخوارزمية "الذكاء الاصطناعي" من الخوارزميات المدمجة المستخدمة في الأمثلة والمfinity على تقنية السويت. وقد استوحيت من استراتيجيات الطائر والقرود التي تتيح للذكاء الاصطناعي مزود قوة في حل المشكلات المعقدة. وتم استخدام تجارب لحل مشاكل محددة من المشكلات المستمرة والمجردة للحل العملية الأمثلة. ومع ذلك، فإن السلبية الرئيسية لهذا تمكن في أنه قد تجمع فيHighlighted Text in Document

未曾提到的英文文献


ملخص البحث:

تعود "الخوارزمية "الذكاء الاصطناعي" من الخوارزميات المدمجة المستخدمة في الأمثلة والمfinity على تقنية السويت. وقد استوحيت من استراتيجيات الطائر والقرود التي تتيح للذكاء الاصطناعي مزود قوة في حل المشكلات المعقدة. وتم استخدام تجارب لحل مشاكل محددة من المشكلات المستمرة والمجردة للحل العملية الأمثلة. ومع ذلك، فإن السلبية الرئيسية لهذا تمكن في أنه قد تجمع فيHighlighted Text in Document

未曾提到的英文文献


ENHANCED ULTRA-WIDE BAND HEXAGONAL PATCH ANTENNA

Yanal S. Al-Faouri, Noor M. Awad and Mohamed K. Abdelazeez

(Received: 14-Aug.-2018, Revised: 17-Sep.-2018, Accepted: 24-Sep.-2018)

ABSTRACT
An enhanced hexagonal shaped planar antenna is presented for ultra-wideband (UWB) applications. In this paper, a hexagonal patch with six circular cuts at its vertices is designed on FR4-substrate with 50 Ω microstrip triangular tapered feed line and a bevelled partial ground plane with five half circular sleeves. The design is investigated using the high-frequency structure simulator (HFSS). The simulated and measured scattering parameter $S_{11}$ (Reflection Coefficient) results show good impedance matching in the frequency range (3 - 27.57 GHz) satisfying return loss ($RL = |S_{11}| \geq 10$ dB with a percentage bandwidth (PBW) of 160.75%. High gain and efficiency, radiation pattern similar to the electric dipole in E-plane and good omnidirectionality in the H-plane are achieved.

KEYWORDS
Ultra-Wide Band (UWB), Reflection coefficient, Circular cuts, Sleeves, Bevel, Gain and bandwidth.

1. INTRODUCTION
The most important requirements in modern communication systems are to provide a wide frequency range with very low power consumption. The UWB wireless technology is launched in 2002 by the Federal Communication Commission (FCC) which authorizes the unlicensed use of the frequency band 3.1 to 10.6 GHz with PBW (PBW = 100% * bandwidth/center frequency) of 91.33% and -41.3 dBm/MHz maximally allowed radiated power [1]. Nowadays, the new wireless communication systems in military and civilian applications are searching for an enhanced wideband that covers both the short and long frequency ranges.

Different techniques have been proposed by different researchers to enhance the antenna bandwidth with a low profile and compact size. Rectangular patch with one round cut in its four corners with one ground groove that has a shape composed of triangle and rectangle to get 8.28 GHz bandwidth (3.42 - 11.7 GHz) with a PBW equal to 109.52% is presented in [2]. Three ground plane modifications consisting of two rectangular sleeves, two rectangular slots and one rectangular groove are introduced in [3] to achieve a bandwidth of 19 GHz (3.4 - 22.4 GHz) with a PBW equal to 147.29%. Two trapezoidal patches are etched on both sides of the substrate with a microstrip feed line to increase the PBW to 114.28% [4]. Adding three steps in the lower patch corners of rectangular shape patch and using microstrip feed line are proposed in [5] to increase the bandwidth to (2.33 - 12.4 GHz) with a PBW equal to 136.73%. Adding a rectangular slit and attaching L- and T- shaped stubs on the radiating circular patch with an offset feed achieve a PBW of 127.87% (3.08 to over 14 GHz) [6]. Cutting a bevel in the rectangular patch and etching two rounded inverted L-shaped slots with an open end in the square ground plane achieve a PBW of 129.18% (2.7 - 12.55 GHz) [7]. A triangular patch with one rectangular slot and two slits fed by coplanar feed line with the deflected ground are used to increase the PBW to 112.5% (2.8 - 10 GHz) [8].

Hexagonal patch antennas are studied by different researchers and achieve good PBW. Folded hexagon UWB patch antenna with an offset feed at one of its vertices achieves a PBW of 144.33% (2.796 - 17.296 GHz) [9]. Hexagonal patch antenna with two symmetrical slots is etched at the center of the patch with a microstrip feed line to achieve a PBW of 109.09% (2 - 6.8 GHz) [10]. A spanner shape hexagonal patch antenna is designed by defecting the patch with a rectangular shape slot to achieve a PBW of 118.79% (2.95 - 11.58 GHz) [11]. A coplanar waveguide (CPW)-fed hexagonal patch antenna

Y. S. Al-Faouri¹, N. M. Awad² and M. K. Abdelazeez³ are with Electrical Engineering Department, The University of Jordan, Amman, Jordan.

Emails: 'y.faouri@ju.edu.jo, 'n.awad@ju.edu.jo and ' Abdelazeez@ieee.org.
with six small hexagonal elements (fractal elements) is added to its corners to achieve a PBW of 93.33% (4 - 11 GHz) [12]. Different configurations of hexagonal shape patch antennas are introduced in [13] with defected ground planes to reduce the antenna size without affecting the bandwidth. One is composed of the hexagonal patch with L-shape and bevel slots and four rectangular slots in the ground plane to achieve a PBW of 120.43% (2.98 - 12 GHz). While the other consists of the hexagonal patch with one horizontal rectangular slot and bevel slots, one circular slot on the feed line and four rectangular slots in the ground plane to achieve a PBW of 135.2% (2.9 - 15 GHz). A new hexagonal patch antenna is proposed in [14] which consists of small five trapezoidal elements which are added to the center of the patch edges, small six hexagonal slots on each of its corners and another hexagonal slot added at the patch center to achieve a PBW of 126.06% (3.1 - 13.67 GHz). Two hexagonal patch antennas are proposed in [15], one of them achieves the UWB with PBW of 129.55% (3.1 - 14.5 GHz) by etching a rectangular groove and cross slot in the ground plane. The other is designed for the super-wideband antenna (SWB) to achieve a PBW of 154.61% (3.2 - 25 GHz) by adding a deep groove that divides the ground plane into two strips laid symmetrically around the feed and triangular slots at the upper corners of the ground plane.

In this paper, a new enhanced hexagonal UWB microstrip antenna design is proposed and investigated. The antenna shape and dimensions are outlined in Section 2. The proposed antenna consists of a triangular tapered microstrip feed line, a hexagonal radiation patch with six circular cuts at the patch vertices and a bevelled partial ground plane with five half circular sleeves. The simulation results and discussions are presented in Section 3. The experimental verifications are outlined in Section 4. Finally, the conclusion is given in Section 5.

2. ANTENNA STRUCTURE

The proposed antenna with the geometrical parameters is shown in Figure 1, where all dimensions are obtained carefully by parametric analysis (explained in Section 3) in order to achieve the desired bandwidth over the needed frequency range. The antenna dimensions (in mm) are: the substrate is FR4 - epoxy with a thickness \( h = 1.6 \), \( \tan \delta = 0.02 \), \( \varepsilon_r = 4.4 \), width \( W_s = 36 \) and length \( L_s = 36 \). A triangular tapered feed line is designed using the equations given in [16] and the length of the triangular tapered feed line is chosen approximately equal to the guided wavelength with \( L_f = 9 \), \( W_f = 3 \) and \( W_t = 1.5 \). The distance between the hexagonal patch and the circular cut edges at the patch middle is \( a = 18 \) and six circular cuts at the vertices with a radius \( b = 2 \). The partial ground plane length \( L_g = 8 \) and width \( W_g = 31 \). The ground bevel has a length \( s = 7 \) and is located at a distance \( p = 3.5 \) from the lower substrate edge. In order to enhance the design, five half circular sleeves are attached to the ground plane in the bottom layer with a radius \( r_s = 0.5 \).

![Figure 1. The hexagonal antenna structure; (a) the basic model, (b) the proposed antenna top layer and (c) the proposed antenna bottom layer.](image)

3. DISCUSSION AND PARAMETRIC STUDY

The design started with the basic model which consists of a simple hexagonal patch fed by a rectangular microstrip feed line with a partial ground plane as shown in Figure 1.
(a) (where \( L_{gb} = 11 \text{ mm}, W_{gb} = 40 \text{ mm}, L_{fb} = 12 \text{ mm}, W_{fb} = 3 \text{ mm} \) and \( a_t = 18 \text{ mm} \)). The achieved PBW for this design is very low with bad impedance matching. The enhancement of the bandwidth is achieved using different means, including changing the length and width of the ground plane, adding sleeves to the partial ground plane, changing the length of the tilted ground plane edge, adding patch circular cuts and adjusting the microstrip feed line shape. The effect of each parameter on the bandwidth (with the scattering parameter \( S_{11} \leq -10 \text{ dB} \)) was performed and optimized to reach the final design. The parametric results in the paper are generated from the final design by varying one parameter at a time and keeping all other parameters constant as listed in Section 2 to show the effect of each parameter alone. The bandwidth enhancement process is as follows:

### 3.1 Ground Plane Modifications

The ground plane dimensions are important parameters in the design of the UWB antennas since the bandwidth depends strongly on them [17]. The simulation results for the scattering parameters \( S_{11} \) using HFSS simulator for \( L_g \) equal to 7, 8 and 9 mm are shown in Figure 2. High values of \( L_g \) cause narrowband with a lot of rejection bands, while decreasing \( L_g \) leads to high bandwidth with lower starting operating frequency. Choosing \( L_g = 8 \text{ mm} \) achieves higher bandwidth, but it still needs to adjust the impedance matching over the different frequency ranges.

![Figure 2. The reflection coefficient when varying the ground length (Lg).](image)

A parametric study is also conducted on the ground plane width \( (W_{gb}) \) between \((29 - 33) \text{ mm}\), where better impedance matching is achieved with lower starting operating frequency when \( W_{gb} = 31 \text{ mm} \). The presence of sleeves in the ground plane increases the inductive part of the input impedance and generates additional resonant mode to improve the overall bandwidth [3, 18]. Parametric analysis is performed on the half circular sleeves’ parameters to investigate their influence on the antenna performance. The sleeve parameters are the sleeve numbers \( (N) \) and their radii \( (r_s) \), where \( r_s \) is varied between \((0 - 1) \text{ mm}\), while \( N \) is between \((3 - 7) \). The simulated reflection coefficient for the sleeve number \( (N) \) variation shows that using \( N = 5 \) achieves lower starting operating point, wider bandwidth and better impedance matching. The reflection coefficient for the sleeve radius variations is shown in Figure 3, where it can be noticed that varying \( r_s \) will affect mainly the impedance matching and consequently the covered bandwidth. The optimum value to use for the proposed antenna is \( r_s = 0.5 \text{ mm} \) and \( N = 5 \).

Bevelling the ground top corners with symmetrical tilted cuts generate more resonant frequencies and adjust the input impedance imaginary part which leads to wider impedance bandwidth [19]-[20]. Parametric analysis is conducted on the ground cut length \( (s) \) between \((5 - 9) \text{ mm}\), where the optimum value for \( s \) is \( 7 \text{ mm} \), since below and beyond this value, the impedance matching degrades at different frequency bands.

### 3.2 Using Triangular Tapered Feed Line

The use of a triangular tapered feed line will enhance the matching between the feeding point and the hexagonal patch to smooth the current path and reduce the incident wave reflection, which achieves
wider bandwidth [16]. The reflection coefficient comparison when using rectangular and triangular tapered feed lines and by keeping all dimensions as in Section 2 is shown in Figure 4. It can be observed that higher bandwidth and better impedance matching are obtained when using a triangular tapered feed line.

Figure 3. The reflection coefficient when varying the sleeves radius ($r_s$) using five sleeves.

Figure 4. The reflection coefficient to compare between the rectangular and triangular tapered feed lines.

3.3 Incorporating Symmetrical Circular Cuts at the Hexagonal Patch Vertices

Adding cuts in the hexagonal patch leads to perturb the surface current path length which will generate more than one resonant frequency and increase the bandwidth. Parametric analysis is conducted to examine the effect of the circular cuts’ parameters on the bandwidth. The important parameters to study are the number of the circular cuts ($M$) and their radius ($b$), where $M$ is varied between (0 - 6) and $b$ between (1 - 3) mm. The location of the cuts is varied such as for $M = 2$ the cuts are on the middle corners and for $M = 4$ the cuts are in the upper and the lower corners, while for $M = 6$ the cuts are on every corner. The simulated results for varying the cuts’ number $M$ are shown in Figure 5, which confirms that increasing the number of cuts will enhance the impedance matching and the covered bandwidth. The simulated reflection coefficient for radius variation is shown in Figure 6. Varying the cut radius $b$, the lower operating frequency changes, because the current path on the patch is perturbed. The main noticeable effect of the patch cuts’ radius $b$ is at the impedance matching. It is concluded that the optimum value for $M$=6 and for $b$ = 2 mm. The proposed antenna reflection coefficient versus frequency using HFSS software tool before and after all modifications is compared in Figure 7. The achieved bandwidth when $RL \geq 10$ dB ranges from 3 till 27.57 GHz with a PBW equal to 160.75% with better impedance matching.
The simulated radiation patterns for the E and H planes at various frequencies 4, 6, 15 and 20 GHz are shown in Figure 8, where the E and H planes are the yz-plane ($\phi = 90^\circ$ and $0^\circ < \theta < 180^\circ$) and the xz-plane ($\phi = 0^\circ$ and $0^\circ < \theta < 180^\circ$), respectively. In the E-plane, the antenna exhibits a dipole shape at low-frequency range, but because of the existence of higher-order modes, the number of lobes rises with the increase of frequency. The H-plane shows good omnidirectionality at low-frequency range and becomes less omnidirectional with an increase in frequency. Figure 9 shows the simulated peak realized gain, where the gain has a low value at the start of the desired band and increases as a function of the frequency.
band and ranges between 5 and greater than 7 dB. The radiation efficiency for the proposed antenna is shown in Figure 10, where it starts at 98% and ends with 80%.

![Radiation Patterns](image1.png)

Figure 8. The radiation patterns at: (a) 4 GHz, (b) 6 GHz, (c) 15 GHz and (d) 20 GHz (E-- & H--).

![Gain Graph](image2.png)

Figure 9. The proposed antenna peak realized gain.

Comparison between different published works and the proposed antenna are shown in Table 1. The proposed antenna achieves a wide impedance bandwidth with simple design compared to other works in [9]-[15]. Although the proposed antenna has a larger substrate area than those in [9], [12]-[13] and [15], the achieved impedance bandwidth is larger.
Table 1. Comparison between the proposed antenna and antennas presented in other published works.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Antenna substrate area (mm²)</th>
<th>Impedance bandwidth (GHz)</th>
<th>Percentage Bandwidth (PB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[9]</td>
<td>27 x 24</td>
<td>2.8 - 17.3</td>
<td>144.33 %</td>
</tr>
<tr>
<td>[10]</td>
<td>54 x 49</td>
<td>2 - 6.8</td>
<td>109.09%</td>
</tr>
<tr>
<td>[11]</td>
<td>31 x 52</td>
<td>2.95 - 11.58</td>
<td>118.79%</td>
</tr>
<tr>
<td>[12]</td>
<td>25 x 25</td>
<td>4 - 11</td>
<td>93.33%</td>
</tr>
<tr>
<td>[13]</td>
<td>13 x 46.5</td>
<td>2.9 - 15</td>
<td>135.2%</td>
</tr>
<tr>
<td>[14]</td>
<td>39 x 36.5</td>
<td>3.1 - 13.67</td>
<td>126.06%</td>
</tr>
<tr>
<td>[15]</td>
<td>35.5 x 30.35</td>
<td>(UWB) 3.1 – 14.5 (SWB) 3.2 - 25</td>
<td>(UWB) 129.55% (SWB) 154.61%</td>
</tr>
<tr>
<td>Proposed Antenna</td>
<td>36 x 36</td>
<td>3 - 27.57</td>
<td>160.57%</td>
</tr>
</tbody>
</table>

4. EXPERIMENTAL VERIFICATION

The designed antenna is fabricated on an FR4 substrate with a dielectric constant $\varepsilon_r = 4.4$ and a height $h = 1.6$ mm as shown in Figure 11. The facilities at King Abdullah Design and Development Bureau (KADDB) were utilized to test the fabricated antenna. The reflection coefficient is measured using Agilent N5242A vector network analyzer. The simulated and measured reflection coefficient results for the proposed antenna are shown in Figure 12 and they compare favourably. The difference between the measured and the simulated results are due to different factors which are not considered through the simulation process, such as the accuracy and precision of fabrication techniques used, the SMA connector welding and its effect in the frequency range beyond 18 GHz, as well as the non-homogeneous behaviour of the FR4 substrate with frequency variations.

5. CONCLUSION

A new planar enhanced UWB antenna is designed for UWB applications. The proposed antenna consists of a hexagonal radiation patch with six circular cuts at its vertices, a triangular tapered microstrip feed line and a bevelled partial ground plane with the addition of five half circular sleeves. The design is investigated using electromagnetic simulators HFSS. The simulation results show good impedance matching over (3 - 27.57) GHz for a return loss (RL) $\geq$ 10 dB. Results of the measurements and simulation compare favourably. Higher gain and efficiency, as well as dipole shape radiation patterns in the E-plane and omni directionality in the H-plane, are obtained.

Figure 11. The fabricated proposed antenna: (a) patch antenna and (b) ground plane.

Figure 12. The simulated and measured reflection coefficient curves for the proposed antenna.

ACKNOWLEDGEMENTS
The authors acknowledge the valuable support of the technical staff in the KADDB through the performed measurements of the proposed antenna.

REFERENCES


ملخص البحث:
في هذه الورقة، يُقدم هواوي مسطح محسن سداسي الشكل من أجل تطبيقات النظام التردد الفائق العلوي. يتم تصميم روعة سداسية بست قنوات دائرية عند رؤوسها على مادة أساس من نوع Cu، مع خط تغذية مثلث مستدقة على شكل شريط صغير مقاسه 50 أوم، ومع سطح أرضي مستو مشكل من زجاجي يحتوي على خمسة أقسام صغرى دائرية. وقد تم استكشاف التصميم المقترح باستخدام مضال بنيوي للترددات S11 عالية، إذ أظهرت النتائج المتعلقة بالمحاكاة والقياس فيما يخص معامل الشعاع، مع رؤوس Cu. وحقق 75.75% من الأداء، وحقق 160.75% من الأداء، مع تحسين في جميع الأدوات في المستوى H، مع إشارة إلى جميع الأدوات في المستوى E.
ENHANCING THE ACCURACY OF SONBOL’S ARABIC ROOT EXTRACTION ALGORITHM

Nisrean Thalji¹, Nik Adilah Hanin¹, Zyad Thalji² and Sohair Al-Hakeem³

(Received: 11-Sep.-2018, Revised: 5-Oct.-2018, Accepted: 25-Oct.-2018)

ABSTRACT

Root extraction is an important primary process in most Arabic applications, such as information retrieval systems, text mining, text classifiers, question answering systems, data compression, indexes, spelling checkers, text summarization and machine translation. Any weaknesses of root extraction will affect negatively the performance of these applications. Sonbol’s Arabic root extraction algorithm achieves high accuracy of performance and gives new classification for Arabic’s letters which minimizes the affix ambiguity. The comparison and testing of the existing Arabic root extraction algorithms on unify datasets shows that they still need some enhancements. Arabic root extraction is mainly based on using patterns, where as much as the algorithm has patterns as much as the accuracy is better. In this study, we improve Sonbol’s Arabic root extraction algorithm, by enhancing its rules and increasing its patterns. We use 4320 patterns to extract the roots, which is the largest patterns’ list extracted by Thalji’s corpus. We test the new algorithm on Thalji’s corpus that contains 720,000 word-root pairs. This corpus is mainly built to test and compare Arabic root extraction algorithms. The new algorithm is compared with Sonbol’s Arabic root extraction algorithm. The algorithm of Sonbol et al. achieves an accuracy of 68%, whereas the new algorithm achieves an accuracy of 92%.

KEYWORDS

Arabic root extraction algorithm, Stemming, Arabic language processing.

1. INTRODUCTION

Arabic language is one of the most used Semitic languages. Semitic languages are spoken in a number of regions that were common in distant times in many regions of Africa and Asia over many decades. Some of these languages are not used now, such as Akkadian, Assyrian and Babylonian and some languages are still used nowadays, such as Arabic, Hebrew and Syriac. Semitic languages are a branch of the Afro-Asiatic language family originating in the Middle East (Bennett, 1998) [1].

In Arabic, vowels are used to ensure the exact meanings of words. If the word is non-vocalized, in many cases it is an ambiguous word and then we need to read the sentence and sometimes the whole text to understand the exact meaning. These vowels are written above or under the letter. Table 1 shows vowels in Arabic and corresponding letter/s in English. Some words in non-vocalized texts may have more than one meaning (ambiguous words). So, they have different roots. For example, the non-vocalized Arabic word /WALEDAYN/ has three possible words /WA ADDAYN/ and /WA ADDEEN/. And then, the possible roots are “/WALAD/” (son), “/DAYN/” (debt) and ”/DEEN/” (religion). Another example is the non-vocalized Arabic word “/KTB/” which has three possible interpretations: “/KUTIBA/” (he wrote), “/KUTIB/” (has been written), and “/KUTUBUN/” (books) [2]. We converted the Arabic letters and words into Latin characters (uppercase), so that the reader who is not familiar with Arabic texts can read it with ease. This way, the reader will be able to read words as the way they sound phonetically.

Root extraction is an important primary process in most Arabic applications, such as information retrieval systems, text mining, text classifiers, question answering systems, data compression, indexes, spelling checkers, text summarization and machine translation.

1. N. Thalji and N. Hanin are with Department of Computer Engineering, School of Computer and Communication Engineering, University Malaysia Perlis, Malaysia. Emails: nntthalji3980@gmail.com, adilahhanin@unimap.edu.my

2. Z. Thalji is with Department of Management Information System, Imam Abdulrahman Bin Faisal University, Kingdom of Saudi Arabia. Email: zythalji@iau.edu.sa

3. S. Al-Hakeem is with Department of Computer Science, Ajloun National University, Jordan. Email: drsosphair@gmail.com
Table 1. Vowels in Arabic and corresponding letter/s in English.

<table>
<thead>
<tr>
<th>No.</th>
<th>Vowels in Arabic</th>
<th>Corresponding letter/s in English</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ء</td>
<td>E</td>
</tr>
<tr>
<td>2</td>
<td>ء</td>
<td>O</td>
</tr>
<tr>
<td>3</td>
<td>ء</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>ء</td>
<td>No letter</td>
</tr>
<tr>
<td>5</td>
<td>ء</td>
<td>En</td>
</tr>
<tr>
<td>6</td>
<td>ء</td>
<td>Un</td>
</tr>
<tr>
<td>7</td>
<td>ء</td>
<td>An</td>
</tr>
<tr>
<td>8</td>
<td>ء</td>
<td>Duplicate the letters</td>
</tr>
</tbody>
</table>

Therefore, many Arabic root extraction algorithms are presented in many different studies that tried to algorithm of Sonbol, Ghneim and Desouki [3]. The algorithm of Sonbol et al. comes after the algorithm of Khoja and Garside [4], which is well-known in extracting Arabic roots. Khoja and Garside algorithm’s accuracy amounted to 95% when they tested their algorithm with their own dataset. Sonbol et al. tried to improve Arabic root extraction algorithms in order to increase the percentage of accuracy. The algorithm of Sonbol et al. accuracy amounted to 98% when they tested their algorithm using their own dataset.

Al-Shawakfa, Al-Badarneh, Shatnawi, Al-Rabab’ah and Bani-Ismail [5] made a comparison study of existing Arabic root extraction algorithms. This comparison included the algorithm of Sonbol et al., Khoja and Garside algorithm and other algorithms. This comparison was conducted in a unified dataset, in order to evaluate these algorithms fairly. Khoja and Garside algorithm’s accuracy was 34%, whereas Sonbol algorithm’s accuracy was 24%. Variance in the accuracy values is due to the differences of datasets that were used in the testing process. The study by Al-Shawakfa et al. revealed that existing Arabic root extraction algorithms still need more improvement. It also presented some weaknesses of Khoja and Garside’s algorithm and the algorithm of Sonbol et al. In this study, we continue completing Sonbol’s work by improving their algorithm.

There are three different approaches to extract the roots of the word; rule-based approach, lookup table approach and statistics-based approach. Recently, most of the root extraction algorithms are rule-based approach [5]. This approach mainly has two parts; the lists of affixes (roots and patterns) and the rules. Each Arabic root extraction algorithm tried to enhance the lists and/or the rules.

However, most algorithms suffer from the following problems:

- There are neither standard dataset nor complete lists of Arabic affixes, patterns and roots. Each work collects or generates their own dataset or lists. Most of the lists which contains the affixes, patterns, roots and lists that they used in each work are not publicly available. They just wrote samples of these lists. As a result, every time a new root extraction method is proposed, the researchers need to collect their own data or generate their own list independently. This will cause overlapped works, where each work is trying to solve the same problem instead of improving each other’s work, which resulted a waste of time and resources. In addition, the lists used might have significant difference in terms of number of words, which will make it difficult for researchers to fairly compare the performance of existing works. Therefore, in recent works, the researchers tried to extend these lists by adding new contents [6].

- Arabic has a complex structure, which makes it difficult to extract the roots [7]. All Arabic root extraction algorithms suffer from affixes’ ambiguity, so that it is difficult to distinguish between affix letters and root letters.

This work focus on solving these problems. The structure of this paper is organized as follows; in Section 2, different related previous studies and their drawbacks are discussed. Section 03 describes the proposed methodology which includes the details of each process. Section 04 explains the experimental implementation of our algorithm and its evaluation process. Section 5 concludes the main points of the paper and gives some future directions.
2. Previous Studies

In this section, we give a brief overview of previous rule-based Arabic root extraction algorithms. Khoja and Garside and Garside algorithm [4] is a very popular rule-based Arabic root extraction algorithm.

Khoja and Garside and Garside algorithm reported 96% accuracy of their stemmer using newspaper text. Al-Shalabi [8] presented an Arabic root extraction algorithm, which is a rule-based algorithm that is used to extract trilateral roots of Arabic words. This algorithm has been tested on a corpus of 72 abstracts 10582 words from the Saudi Arabian National Computer Conference, where its accuracy was about 92%.

Al-Kabi and Al-Mustafa algorithm [9] is based on affix removal. They tested their algorithm on small data sets containing 1,827 words. The system failed to analyze 55 words, since their patterns are unknown. This failure was mostly due to foreign (Arabized) words. The system enables to analyze the rest 1,772 words and achieved 91% of accuracy.

Sonbol, Ghneim and Desouki [3] algorithm is a rule-based root-extraction algorithm, the principal idea of which is based on encoding Arabic letters with a new code that preserves morphologically useful information and simplifies its capturing toward retrieving the root. They conducted their experiments using two different corpuses. The first corpus consisted of lists of word-root pairs 167,162 pairs. The second corpus was a collection of 585 Arabic articles from different categories (policy, economy, culture, science and technology and sport). This corpus consisted of 377,793 words. In general, the accuracy was about 96%-98%.

Another work is Ghwanmeh, Al-Shalabi, Kanaan, Khanfar and Rabab’ah algorithm [10], which is a rule-based algorithm used to find trilateral Arabic roots. According to Ghwanmeh et al., their algorithm has only failed to analyze words that are normally foreign, irregular or do not have trilateral roots. A corpus of 242 abstracts from the Proceedings of Saudi Arabian National Computer conference in machine-readable form was used in the testing procedure. The set of abstracts was chosen randomly from the corpus for analysis. The results obtained showed that the algorithm extracts the correct roots with an accuracy rate up to 95%. Many algorithms have been conducted under this type, like the Kchaou and Kanoun algorithm [11], El-Defrawy, El-Sonbaty and Belal algorithm [12] and Ayedh and Guanzheng algorithm [13].

Also, many morphological analyzers have been conducted to properly provide maximum morphological information on Arabic words, such as the proclitic, the prefix, the lemma, the suffix, the stem, the root, the enclitic, the tag and the pattern. One of them is MADAMIRA [14], a morphological analyzer that provides many information on Arabic words. MADAMIRA combines two morphological analysis systems; MADA [15], [16] and [17] and AMIRA [18].

In addition, Al-Khalil Morphological System 2 [19] is a recent morphological analyzer that provides many information on Arabic words. It deals with vocalized and non-vocalized Arabic words. It overcomes many errors of the previous system Al-Khalil Morphological System 1.

In general, all rule-based Arabic roots extraction algorithms share seven processes, which are: normalization, removing prefixes and suffixes, matching the word against patterns, extracting the roots from the patterns, comparing the roots with the roots’ list, returning the extracted roots and finally making enhancement to extract the correct roots, as shown in Figure 1.

The difference between one algorithm and the others is in the details of each process. Also, every algorithm has different lists of prefixes, suffixes, roots and patterns. The main problem is in process two and process four. In process two, in many cases, the algorithms remove the matched prefixes and suffix letters, but these letters are part of the root. So, the result is a wrong root. Our proposed solution to this problem is done by not removing prefixes or suffixes and collecting more rules to reduce affix ambiguity. In process four, the algorithms match the word against patterns to extract the root. The main problem is the limited number of patterns that are collected till now. The extraction accuracy will improve if the algorithm can test as many as existing word patterns. Our proposed solution to this problem is using longer pattern lists. The proposed algorithm uses Thalji’s pattern [6], which is the most recent list. This list is automatically generated from most of the Arabic dictionaries and contains (4320) patterns, which is the longest list discovered until now. These patterns contribute to enhancing the accuracy of Arabic root extraction algorithms.
Enhancing the Accuracy of Sonbol’s Arabic Root Extraction Algorithm, N. Thalji, N. Hanin, Z. Thalji and S. Al-Hakeem.

3. Method

The root is the base form of the word that gives the main meaning of the word. In this section, the methodology for the proposed Arabic root extraction algorithm is explained. The proposed algorithm is an enhancement of the algorithm of Sonbol et al. by increasing the rules and the lists to find all possible roots of the word.

3.1 Normalization

The normalization steps for the algorithm of Sonbol et al. are as follows:
- Removing the kasheeda symbol ("_ ").
- Removing the diacritics.
- Replacing the Hamza’s forms (ء، آ، ؤ، إ، ئ) with the letter (‘).

In this section, we extend Sonbol’s normalization process by the following steps:
- Removing the punctuations.
- Removing the non-letters.
- Duplicating any letter that has the Shaddah: “َّ” symbol.

3.2 Encoding

In this step, Sonbol coded the Arabic letters based on six symbols {O, P, S, PS, U, A}, representing six groups of letters each of which shares certain characteristics:

**O:** Original letters. These letters are surely part of the root. They are: (“ث”, “THAA”), (“ج”, “JEEM”), (“ح”, “HAA”), (“خ”, “KHAA”), (“د”, “DAL”), (“ز”, “THAL”), (“ر”, “RAA”), (“ز”, “ZAY”), (“ش”, “SHEEN”), (“ص”, “SAD”), (“ث”, “THAD”), (“ط”, “DAA”), (“ث”, “TAA”), (“غ”, “AYEN”), (“غ”, “GAF”), (“غ”, “GAYEN”). This means that if the word contains one or more of these letters, these letters should be part of root’s letters.

**P:** Prefix letters (ب، س، ف، ب) (BAA, FAA, SEEN, LAM). These letters should be treated as part of the root word if they appear in a different part of the word other than the prefix part. Otherwise, these letters are considered ambiguous letters (can be part of the root word or added letters to the root word). If these letters are ambiguous letters, the algorithm initially considers them as prefix letters. There is a possibility for these letters to become (O) letters (root’s letters) in the next steps.

**S:** Suffix letter (“ه”, Haa). This letter should be treated as part of the root word if it appears in a different part of the word other than the suffix part. Otherwise, this letter is considered as an ambiguous letter. If this letter is an ambiguous letter, the algorithm initially considers it as suffix letter. There is a possibility for this letter to become (O) letter (root’s letter) in the next steps.

**PS:** Prefix-Suffix letters (م، م، ن) (KAF, MEEM, NOON). These letters can appear only on both sides of the word; i.e., in the suffix part or in the prefix part. These letters should be treated as part of the root

---

Figure 1. Main processes in rule-based Arabic root extraction algorithms.
word if they appear in a different part of the word other than the prefix-suffix part. Otherwise, these letters are considered ambiguous (can be part of the root word or added letters to the root word). If these letters are ambiguous letters, the algorithm initially considers them as prefix-suffix letters. There is a possibility for these letters to become (O) letters (root’s letters), (P) prefix letters or (S) suffix letters in the next steps.

U: Uncertain letters ("أ، و، ي، ت") (TAA, YAA, WAW, ALEF)). These letters can appear anywhere in the word. It is not possible to verify whether these letters are part of the root word letters. Several cases are associated with these letters as they may change, omit or convert from one letter to another during the derivation process following well known Arabic rules “EBDAL and EALAL”. For example, the Arabic word “فل KEEL” (It was said) and its root “قول”，the letter “و” is converted into “ي” during the derivation process.

A: Added letter (“ُ” only (TAA MARBUTA)). This letter is always considered an additional letter. This letter is always deleted.

3.3 Some Improvements to the Last Coding

In this section, Sonbol et al. added some improvements to the last coding, by applying the following conditions:

- The letter “ب” (BAA) is a prefix letter if it is situated in the first three letters; otherwise, it is an original letter (part of the root word letter). This rule means that if the letter “ب” (BAA) is situated among the first three letters, it is an ambiguous letter. On the other hand, it will be part of the root word if it appears in place other than the first three letters of the word. For example, with the words “بامر雷锋"(BEAMRK, WBLSA"EED), the letter “ب” (BAA) is situated among the first two letters. So, it’s an ambiguous letter. Initially, the algorithm considers it as a prefix letter. Then in the next steps, it may change to (O) letter (root’s letter). Another example is the word "بفايوباتل "(AFBLBATL); this word has the letter “ب” (BAA) appearing twice in the word. The first “ب” (BAA) letter is an ambiguous letter, as it appears as the third letter in the word. Initially, the algorithm considers it as a prefix letter. Then, in the next steps, it may change to (O) letter (root’s letter). The second “ب” (BAA) will be considered part of the root word, as it appears as the sixth letter of the word.

- The letter “ف” (FAA) is a prefix letter if it appears among the first two letters; otherwise, it is an original letter. This means that if the letter “ف” (FAA) appears among the first two letters of the word, it is an ambiguous letter. If the letter “ف” (FAA) appears in place other than the first two letters, it is part of the root word. Initially, the algorithm considers it as prefix letter. For example, in the words "ففارسأجراتها الف" (FHD/FASTAIRTHA, AFGYR), the letter “ف” (FAA) appears among the first two letters of the word. So, it’s an ambiguous letter. Initially, the algorithm considers it as a prefix letter. It may change to (O) letter (root’s letter) in the next steps of the algorithm following certain rules. Another example; the words "ف "(TSAFHT), "ف" /MTFAELEEN and "ف "/WETFKFT; the letter “ف” (FAA) is considered to be part of the root word as it appears in places other than the first two letters of the word.

- The letter “س” (SEEN) is a prefix letter if it is followed by one of the letters "," (HAMZA, NOON, YAA AND TAA); otherwise, it is part of the root word. For example, the words "سستفي" /STBAKA, "ستنواتج" /SATGYR, "ستنواتج" /SNTWAJD, "ستنواتج" /SYNILE; the letter "س" (SEEN) is an ambiguous letter, as it is preceding one of the letters "," (HAMZA, NOON, YAA AND TAA). Initially, the algorithm considers it as a prefix letter. It may change to (O) letter (root’s letter) in the next steps of the algorithm. Another example is with the words "كالاسد" /KALASD, "مسور" /MASOR; the letter "س" (SEEN) is considered part of the root word, as it is not preceding one of the letters "," (HAMZA, NOON, YAA AND TAA).

- The letter “ل” (LAM) is considered a prefix letter if it appears among the first five letters of the word; otherwise, it is part of the root word.

- The letter “ه” (HAA) is considered a suffix letter if it appears among the last three letters of the word; otherwise, it is part of the root word.

- The letter “ك” (Kaf) is considered a prefix letter if it appears among the first three letters of the word; otherwise, it is a suffix letter.
If the algorithm of Sonbol et al. finds three O-Letters (or more) in the encoded word, these letters are considered root letters and the algorithm is terminated. However, in this work, the enhancement of the algorithm of Sonbol et al. is to continue searching for other possible roots and for longer roots (more than three letter root word).

3.4 Applying Transformation Rules

In this section, the algorithm of Sonbol et al. applies transformation rules between groups to obtain a maximum number of original letters. Transformation rules are mentioned below:

- R1) Change each (P) after (O) to (O).
  
  For example, with the word “ضـ/DYUF”, “ضـ” (DAA) is an (O) letter, “فـ” (FAA) is a (P) letter; in this case, (P) comes after (O). So, “فـ” (FAA) is changed to (O) letter, which means that it should be one of the root’s letters. Until now “ضـ/ (FAA, DAA) are part of the root’s letters.

- R2) Change each (S) before (O) to (O).
  
  For example, with the word “فـ/ALHDAYH”, “فـ” (DAA) is an (O) letter, “ـبـ” is a (S) letter; in this case, (S) comes before (O). So, “ـبـ/HAA” is changed to (O) letter, which means that it should be one of the root’s letters. Until now “ـبـ/ (FAA, DAA) are part of the root’s letters.

- R3) Change each (PS) before (P) to (P).
  
  For example, with the word “كـ/KAA” is a (PS) letter, “ـبـ/SEEN” is a (P) letter; in this case, (PS) comes before (P). So, “ـبـ/KAA” is changed to (P) letter, which means that it should be one of the root’s letters or prefix letters, but not a suffix letter.

- R4) Change each (PS) before (O) to (P).
  
  For example, with the word “ـبـ/MNTKMON”, “ـبـ” (DAA) is an (O) letter; “ـبـ/KAA” is a (P) letter; in this case, (PS) comes before (O). So, “ـبـ/NOON” is changed to (P) letter, which means that it should be one of the root’s letters or prefix letters, but surely not a suffix letter.

- R5) Change each (PS) after (S) to (S).
  
  For example, with the word “ـبـ/MNTKHM”, “ـبـ/KAF” is a (PS) letter; “ـبـ/HAA” is a (S) letter; in this case, (PS) comes after (S). So, “ـبـ/KAF” is changed to (S) letter, which means that it should be one of the root’s letters or suffix letters, but surely not a prefix letter.

- R6) Change each (PS) after (O) to (S).
  
  For example, with the word “ـبـ/ELMKH”, “ـبـ” is a (PS) letter, “ـبـ/A’A” is an (O) letter; in this case, (PS) comes after (O). So, “ـبـ/KAF” is changed to (S) letter, which means that it should be one of the root’s letters or suffix letters, but surely not a prefix letter.

- R7) Change each (P) after (S) to (O).
  
  For example, with the word “ـبـ/ELTHBT”, “ـبـ” (BAA) is a (P) letter; “ـبـ” (HAA) is an (S) letter; in this case, (P) comes after (S). So, “ـبـ/BAA” is changed to (O) letter, which means that it should be one of the root’s letters. Until now, “ـبـ” (BAA, HAA) are root’s letters.

- R8) Change each (S) before (P) to (O).
  
  For example, with the word “ـبـ/ALBHTAN”, “ـبـ” (S) is an (S) letter; “ـبـ/BAA” is a (P) letter; in this case, (S) comes before (P). So, “ـبـ/BAA” is changed to (O) letter, which means that it should be one of root’s letters. Until now, “ـبـ” (BAA, HAA) are root’s letters.

As in the previous step, if the algorithm of Sonbol et al. has three O-Letters in the encoded word, these letters are considered root letters and the process will terminate here. However, the enhancement of the algorithm of Sonbol et al. is to continue searching for other possible roots and for longer roots, with more than three root length.

3.5 Extracting All Possible Patterns of the Word

The algorithm of Sonbol et al. uses the idea of traditional algorithms, but with the aid of the encoded
word. Traditional algorithms store lists of Arabic prefixes, suffixes and patterns. These algorithms delete prefixes and suffixes, then use the pattern to extract the root from the reminder. The enhancement of this process is done by using larger lists’ content and not removing the prefixes or suffixes, but applying the patterns. It’s worth to mention that the presented algorithm assumes that patterns are composed of three elements: prefix, stem and suffix. One of the problems that were experienced by most of the previous algorithms is that they delete clitics before comparing with patterns. And in many cases, these clitics are parts of roots and not clitics. For example, with the words “الكحل، ل ح” (AL, KAL) will give these roots ح (TKA, HON), ignoring other possible roots “الكحل، ل ح” (LKA, KLH). In this section, we use the pattern’s list of Thalji’s corpus that was automatically extracted from most well-known Arabic dictionaries. It is the largest list of 4,320 patterns, which is the most appropriate list to be used in this work. Previous algorithms have used short lists that were manually collected. In addition, they did not publicly publish all the lists’ contents. Thalji’s patterns are listed in appendix A, so that future researchers can benefit from them.

In this step, we compare the word with the Thalji’s list of patterns and return all matched patterns. For example, for a word like “ف ه د/ FHD”, the algorithm found two original letters, “د/ DAA, HAA”. Next, the word is compared to the list of patterns and all matched patterns were returned. The word “ف ه د/ FHD” matches the pattern”ل/ FA’L”. The word “ف ه د/ FHD” is the first possible root. Another example is the word “الب/ ALBAHER”, where the algorithm just finds two original letters, which are”ر/ RAA, HAA”. The word is compared to the list of patterns and all matched patterns were returned. The word “الب/ ALBHR” matches the pattern “الب/ ALFA’L”. The word “الب/ ALBHR” is the first possible root. Also, this word matches the patterns “الب/ ALBHR/ EFA’L”, then “الب/ ALBHR/ ABHR” are also possible roots.

### 3.6 Extracting All Possible Roots for the Word

All possible roots are found by matching the words against the list patterns. All the possible roots that match the patterns are extracted after finding all possible patterns.

### 3.7 Solving the Problems with Ealal Rules and Ebdal Rules

When we have a weak letter (ALEF, YAA AND WAW), we replace this letter with the two other letters and check if the result is a valid root. If so, we add this root to the possible roots. For example, in the word “ف ه د/ KAL”, the algorithm replaces “/” ALEF” with “ي/ YAA” and “و/ WAW”. So, “ق ي ذ، ف ه د/ ق ي ذ” (KEEL, KAWL) are possible roots.

### 3.8 Minimizing Possible Roots by Comparing them with Roots’ List

In this section, we use the roots’ list of Thalji’s corpus that was automatically extracted from most well-known Arabic dictionaries. It is the largest roots’ list found till now with 12,000 roots. This list is longer than the list that is used in Ababneh, Al-Shalabi, Kanaan and Al-Nobani stemmer [20]. It has about 11,347 roots. The distribution of roots for these two different lists is shown in Table 2.

<table>
<thead>
<tr>
<th>Roots</th>
<th>List of Thalji’s corpus</th>
<th>List of Ababneh, Al-Shalabi, Kanaan and Al-Nobani stemmer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-letter roots</td>
<td>500</td>
<td>115</td>
</tr>
<tr>
<td>Three-letter roots</td>
<td>7912</td>
<td>7198</td>
</tr>
<tr>
<td>Four-letter roots</td>
<td>3180</td>
<td>3739</td>
</tr>
<tr>
<td>Five-letter roots</td>
<td>360</td>
<td>295</td>
</tr>
<tr>
<td>Six-letter roots</td>
<td>48</td>
<td>0</td>
</tr>
</tbody>
</table>
The presented algorithm uses Thalji’s list to minimize the possible roots, whereas the algorithm of Sonbol et al. used a short list of roots. For example, in the word “البحر” (BHR, ALBHR, ABHR), the possible roots are “اب,ر” (ALBHR, ABHR), while the roots “اب,ر, الب,ر” (BHR, ALBHR, ABHR) are excluded, because they are not found in the roots’ list.

4. EXPERIMENT AND EVALUATION

In this section, the presented algorithm is compared with other algorithms with the same approach, which is the rule-based approach. These algorithms are Khoja and Garside’s Arabic root extraction algorithm and Sonbol’s Arabic root extraction algorithm. In addition, the presented algorithm is also compared with one of the most recent morphological analyzer systems, which is Al-Khalil Morphological System 2. Al-Khalil Morphological System 2 gives maximum morphological information of Arabic words, such as the proclitic, the prefix, the lemma, the suffix, the stem, the root, the enclitic, the tag and the pattern. A complete comparison was conducted between the algorithms on Thalji’s corpus in terms of accuracy. Thalji’s corpus is an automatic corpus that is built from ten old Arabic dictionaries; this corpus is mainly built to test and fairly compare Arabic root extraction algorithms. This corpus contains 720,000 word-root pairs, which helps to avoid the interference of a human expert normally needed to verify the correct roots of each word used in the testing or comparison process. Moreover, this corpus has more than 4,320 types of words derived from 12,000 roots. Therefore, the list used in this experiment is more comprehensive compared to previous works.

The result of testing shows that the accuracy of Khoja and Garside’s algorithm was 63%, the accuracy of the algorithm of Sonbol et al. was 68%, the accuracy of Al-Khalil Morphological System 2 was 75%, whereas the accuracy of the presented algorithm was 92%. Figure 2 shows the performance accuracy of all compared algorithms.

The main problems of Khoja and Garside’s algorithm are that it does not consider many roots, prefixes, suffixes and patterns. It suffered from suffix ambiguity problems. In addition, it returned just one solution for non-vocalized words, ignoring other possible solutions. Besides, it replaced a vowel letter with the letter “و” that sometimes returns a root that is not related to the derivation word. Finally, it produced wrong roots being unsuccessful to extract roots for derivation words that contain the “ابدال” rule.

The main problems of the algorithm of Sonbol et al. arise if the root does not contain any constant letter, if the root does not start with a constant letter or if the root contains only one constant letter. Also, it does not consider many roots, prefixes, suffixes and patterns. In addition, it returned just one solution for non-vocalized words, ignoring other possible solutions.

The main problems of Al-Khalil Morphological System 2 are that it failed to analyze some words, which were about 25% of the input words. Table 3 shows a sample of these words. For example, the words
In future, the presented Arabic root extraction algorithm, Sonbol’s Arabic root extraction algorithm and Al-Khalil Morphological System 2. The testing and comparing processes are conducted on Thalji’s corpus, where the result of testing shows that the accuracy of Al-Khalil Morphological System 2 was 75%. The presented algorithm achieved an accuracy of 92%.

In future, we plan to enhance the accuracy of the presented algorithm, overcome some weakness points and enhance the result to return just the exact root word. In order to implement this, the system must have the ability to understand the whole sentence or sometimes the whole paragraph.
"Enhancing the Accuracy of Sonbol's Arabic Root Extraction Algorithm", N. Thalji, N. Hanin, Z. Thalji and S. Al-Hakeem.

REFERENCES

APPENDIX

Failing the verb below by applying the morpho-syllable and syllable rule to Arabic, a Persian stem Web, and the rest of its root.


ملخص البحث:

إن استخراج الجذور عملية أساسية مهمة في غالبية تطبيقات اللغة العربية، مثل: أنظمة استرجاع المعلومات، وتحليل النصوص، وتصنيف النصوص، وأنظمة الإجابة عن الأسئلة، وضغط البيانات، والترجمة الآلية، وتلخيص النصوص، والنصوص العربية. وإن أي ضعف في استخراج جذور الكلمات من شأنه أن يؤثر سلباً على الأداء في تلك التطبيقات.

تحقيق خوارزمية سُنْنَل لاستخراج الجذور دقيقة عالية في الأداء وتقديم تصنيفياً جديداً للأحرف العربية من شأنه أن يقلل كثيراً من الخصائص المتعلقة بالبادئات واللاحقات. وتثبت المقارنة والعناصر لخوارزميات المتاحة لاستخراج الجذور في اللغة العربية أن تلك الخوارزميات لا تزال في حاجة إلى بعض التحسينات.

يعتمد استخراج الجذور في اللغة العربية على استخدام الأوزان، إذ تزداد دقة العملية بزيادة عدد الأوزان. في هذه الدراسة، يجري تحسين خوارزمية سُنْنَل لاستخراج الجذور من خلال تحسين قواعدها وزيادة أوزانها. وتم استخدام 4320 وزن لاستخراج الجذور، وتم العد عدد قائمة أوزان تم استخدامها من مجموعة قواعد بيانات "ثليج". كما تم فحص الخوارزمية الجديدة باستخدام تلك المجموعة من قواعد البيانات التي تحتوي على 20000 رمز من الجذور والكلمات. والجدید بالذكر أن مجموعة قواعد البيانات المذكورة فيما نتىت لفحص خوارزميات استخراج الجذور ومقارنتها.

لقد تمت مقارنة الخوارزمية الجديدة بخوارزمية سُنْنَل لاستخراج الجذور، وكانت النتيجة أن خوارزمية سُنْنَل حققت دقة بلغت 68%، في حين بلغت دقة الخوارزمية المحدثة الجديدة 92%.

"Enhancing the Accuracy of Sonbol’s Arabic Root Extraction Algorithm", N. Thalji, N. Hanin, Z. Thalji and S. Al-Hakeem.
FEATURE PRUNING METHOD FOR HIDDEN MARKOV MODEL-BASED ANOMALY DETECTION: A COMPARISON OF PERFORMANCE

Sulaiman Alhaidari and Mohamed Zohdy

(Received: 10-Oct.-2018, Accepted: 10-Nov-2018)

ABSTRACT

Selecting effective and significant features for Hidden Markov Model (HMM) is very important for detecting anomalies in databases. The goal of this research is to identify the most salient and important features in building HMM. In order to improve the performance of HMM, an approach of feature pruning is proposed. This approach is effective in detecting and classifying anomalies, very simple and easily implemented. Also, it is able to reduce computational complexity and time without compromising the model accuracy. In this work, the proposed approach is applied to NSL-KDD (the new version of KDD Cup 99), DDoS, IoT POT and UNSW_NB15 data sets. Those data sets are used to perform a comparative study that involves full feature set and a subset of significant features. The experimental results show better performance in terms of efficiency and providing higher accuracy and lower false positive rate with reduced number of features, as well as eliminating irrelevant redundant or noisy features.

KEYWORDS

Anomaly detection; Feature pruning; Hidden Markov Model; NSL-KDD; DDoS; UNSW_NB15; IoT POT.

1. INTRODUCTION

Increasing size of data and network traffic has made information security more complex and challenging than at any time in the past. Information security is intended to protect information (data) and information systems from any malicious activities and unauthorized access [1]-[9]. However, the increasing size of data maximizes number of computations and minimizes detection accuracy rates [9]. Therefore, in recent years, many researchers have worked on this problem and applied the feature pruning method on several data sets to improve the detection performance and obtain faster and more cost-effective results. Using a feature pruning method for machine learning is a way to solve this problem.

This research examines the full feature set on NSL-KDD, DDoS, IoT POT and UNSW_NB15 to reduce the number of features and identify the most salient and significant features that give higher accuracy and lower false positive rate. Therefore, a subset of significant features in detecting anomalies can be obtained by using one of the most popular machine learning techniques, HMM. These significant features can then be used in building an HMM that can effectively achieve much better prediction and detection performance. The remainder of this paper is organized as follows: Following the introduction to our work in Section 1, Section 2 reviews HMM and its problems. Section 3 describes the approach, followed by Section 4 which illustrates the experiments of this study. At the end, conclusions are presented in Section 5.

2. HIDDEN MARKOV MODELS

HMMs have been extensively utilized in many applications, such as speech recognition, finance, computer vision and bioinformatics. Hidden Markov Models can be defined as a tool for representing different probability distributions over sequences of observed variables [1]. In HMM, the sequence of observations is $O = \{O_1, O_2, ..., O_T\}$, where $O_T$ is one of the observations symbols and $T$ is the number of observations in the sequence. This sequence that goes through over time is observable and it’s...
generated by a stochastic process. In contrast, the sequence of states \( X = \{X_1, X_2 \ldots X_N\} \), where \( N \) the number of states in the model, is not visible to the observer; therefore, it’s not directly observed in this process. In a Markov chain, the probability of each state at time \( t \) is predicted based only on the previous state at time \( t-1 \) as shown in Equation (1). Figure 1 shows the first-order Hidden Markov model.

\[
P = (X_{t+1}|X_1 \ldots X_N) = P(X_{t+1}|X_t)
\]

Figure 1. First-order hidden Markov model.

Mathematically, an HMM is defined as:

\[
\lambda = (A, B, \pi)
\]

where,

\( A \) is the state transition probability distribution. It’s a single matrix \( N \times N \) (\( N \) is the number of states), with each element \( a_{ij} \) representing the probability transitioning from state \( X_{t-1}, i \) to \( X_t, j \) as shown in Figure 2. It can be written as:

\[
a_{ij} = P(X_{t-1}, j = 1|X_t, i = 1)
\]

\( B \) is the emission probability. It is a single matrix \( N \times M \) (\( N \) is the number of states and \( M \) is the number of emissions), where each element \( b_{kj} \) represents the probability of making observation \( O_t, k \) given state \( X_t, j \) as shown in Figure 2. It can be written as:

\[
b_{kj} = P(O_t = K|X_t = i)
\]

\( \pi \) is the initial state distribution. I is an initial vector that contains the probabilities of starting in each of the states. It can be written as:

\[
P(X_1|\pi) = \prod_{i=1}^{N} \pi_i
\]
Thus, an HMM can be characterized by a set of three parameters: the probability distribution of the initial states (\(\pi\)), the probability matrix of transition probabilities between states (\(A\)) and the probability matrix of emission probabilities for each state (\(B\)). HMM can be used in real-world applications by solving the following three fundamental problems:

- **The Evaluation Problem:** Given an HMM \(\lambda = (A, B, \pi)\) and a sequence of observations \(O = \{O_1, O_2, \ldots, O_T\}\), compute the probability (likelihood) of the observed sequence that was generated from the system given the model \(P = (O|\lambda)\). This problem can be solved by two algorithms; namely, the forward algorithm and the backward algorithm.

- **The Decoding Problem:** Given an HMM \(\lambda = (A, B, \pi)\) and a sequence of observations \(O = \{O_1, O_2, \ldots, O_T\}\), find the most likely hidden state sequence \(X_1, X_2, \ldots, X_N\) that produced the observed sequence. This problem is solved using Viterbi algorithm.

- **The Learning Problem:** Given an HMM \((\lambda = (A, B, \pi)\) and a sequence of observations \(O = \{O_1, O_2, \ldots, O_T\}\), adjust the model parameters \((A, B, \pi)\) to maximize \(P = (O|\lambda)\) to obtain the most descriptive model for the system. The Baum-Welch algorithm is used to solve this problem.

More expansion of these problems mentioned above and their solutions can be found in [2]

### 3. Feature Pruning Method

Feature pruning method is a method of eliminating features from the original dataset to obtain a subset of features. Reducing features from the full data set will not only reduce the data size that is required to process and the computational complexity, but selecting a good feature set also helps to improve the efficiency and accuracy of the detection model approach. It plays a key role in building detection models. On the other hand, using all features without applying feature pruning method might increase the overhead of the model, which leads to increase the time to build the model [7]. In this study, we present a feature pruning method which was built to be used to choose certain features from a given feature set. Our aim here is to find the significant feature set that gives the highest accuracy. In order to perform feature-pruning method, we first need to standardize the data to a normal distribution and then combine the standardized data to one sequence of observation, which then could then be used afterward with Viterbi algorithm to compute the most likely state sequence and then be compared to the actual sequence of states to determine the accuracy of the feature. The feature pruning algorithm that we implemented automates this process, eliminates each feature from the full set of the features and then checks the accuracy of the subset of features. More features that are least significant are eliminated if the obtained accuracy is within a certain tolerance of the accuracy, equal or higher than the previous accuracy of every feature combined. This process continues until no improvement of the accuracy is observed on elimination of features.

### 4. Experiments

#### 4.1 Method of Performance Testing

To evaluate the performance of our proposed model and how accurate the model classifies and predicts the class label of attack and non-attack, we need to know the following four terms: True Positive (TP): the number of attack instances classified as attacks; True Negative (TN): the number of non-attack instances classified as non-attacks; False Negative (FN): the number of attack instances classified as non-attacks; False Positive (FP): the number of non-attack instances classified as attacks. For this study, we used the following performance measures to test the performance of the proposed model:

**Accuracy:** It is the ratio of the total number of correctly predicted instances to the total number of all instances. In our study, accuracy is measured by using the Viterbi algorithm to generate a likely state sequence and compare it to the known state sequence to get TP, FP, FN and TN. The accuracy can be calculated by using the following equation:

\[
Accuracy = \frac{TP + TN}{TP + TN + FP + FN}
\]  

(6)

**Error rate:** It is the ratio of the total number of misclassifications to the total number of all predictions.
Algorithm 1. Feature pruning algorithm for HMM.

1: Begin  
2: feature_pruning (X, L, Already_Cheched)  
3: Input: X-The number of desired features  
       L-List of N feature Vectors  
       Already_Cheched- A set containing already checked combinations of features  
4: Output: R-The reduced feature set of length X  
5: if Already_Cheched contains (L) then  
6: Return  
7: end  
8: if N-X=0 then //L contains the desired of features so test accuracy  
9: Return L, evaluate accuracy ()  
10: else  
11: A_i= {All features in L except feature i} for all i=1,…,N //Create subsets of L with one less feature  
12: Return max (feature_pruning (X,A_1), feature_pruning (X,A_2), …, feature_pruning (X,A_N)) //Return the subset with best accuracy  
13: end  
14: End

The error rate can be calculated by using the following equation:

\[
\text{Error rate} = \frac{FP + FN}{TP + FP + TN + FN} \tag{7}
\]

Fall-out: It is the ratio of the number of detected false positives to the total number of predictions. The fall-out can be calculated by using the following equation:

\[
\text{Fall out} = \frac{FP}{FP + TN} \tag{8}
\]

Sensitivity: It is the ratio of the total number of detected true positive instances that are correctly identified as attacks to the total number of positive instances. Sensitivity can be calculated by using the following equation:

\[
\text{Sensitivity} = \frac{TP}{TP + FN} \tag{9}
\]

Specificity: It is the ratio of the total number of detected true negative instances that are correctly identified as non-attacks to all the negative instances. Specificity can be calculated by using the following equation:

\[
\text{Specificity} = \frac{TN}{TN + FP} \tag{10}
\]

Precision and recall: Precision and recall measures are widely used for performance evaluation of machine-learning classification methods. Precision is the ratio of the total number of positive instances that are correctly identified as attacks to the total number of attacks. Recall is the ratio of the total number of instances that are correctly identified as attacks to the total number of all the instances that are correctly identified as attacks and misidentified attacks (it is the same as the sensitivity). Precision and recall can be calculated by using the following equations:
\[ \text{Precision} = \frac{TP}{TP + FP} \]  
\[ \text{Recall} = \frac{TP}{TP + FN} \]  

F-measure: F-measure is a testing score that tests the accuracy of the model and considers both precision and recall. F-measure can be calculated by using the following equation:

\[ F - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \]

4.2 Datasets and Discussion

4.2.1 NSL-KDD Dataset

The NSL-KDD 2009 dataset is a revision of the KDD’99 dataset that is extracted from the KDD’99 dataset to solve some of the inherent problems [3]. The size of NSL-KDD dataset is smaller than that of the original KDD’99. In each record of the set, there are 41 different features of the flow and one more attribute for class assigned to each record as either an attack type or a normal type as shown in Table 1. The NSL-KDD dataset contains 23 types of attack in the training set and 17 additional attack types in the testing set (New attacks that are not included in the training set) that are classified into four major categories: DoS, Probe, R2L and U2R, as shown in Table 2. The features in bold in Table 1 are the significant features obtained by the feature pruning algorithm used in our experiment on NSL-KDD dataset. Note: some features such as IP addresses, protocol type and source/destination port numbers were ignored from the initial feature set to guarantee that the results of the detection model are not dependent on particular acquisition biases. Table 2 summarizes the results of the experiment and Figure 3 shows the results in a graphical way.

Table 1. List of features of NSL-KDD dataset.

<table>
<thead>
<tr>
<th>No.</th>
<th>Feature Name</th>
<th>No.</th>
<th>Feature Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Duration</td>
<td>22</td>
<td>is_guest_login</td>
</tr>
<tr>
<td>2</td>
<td>protocol type</td>
<td>23</td>
<td>count</td>
</tr>
<tr>
<td>3</td>
<td>service</td>
<td>24</td>
<td>srv_count</td>
</tr>
<tr>
<td>4</td>
<td>flag</td>
<td>25</td>
<td>serror_rate</td>
</tr>
<tr>
<td>5</td>
<td>src_bytes</td>
<td>26</td>
<td>srv_serror_rate</td>
</tr>
<tr>
<td>6</td>
<td>dst_bytes</td>
<td>27</td>
<td>rerror_rate</td>
</tr>
<tr>
<td>7</td>
<td>land</td>
<td>28</td>
<td>srv_rerror_rate</td>
</tr>
<tr>
<td>8</td>
<td>wrong_fragment</td>
<td>29</td>
<td>same_srv_rate</td>
</tr>
<tr>
<td>9</td>
<td>urgent</td>
<td>30</td>
<td>diff_srv_rate</td>
</tr>
<tr>
<td>10</td>
<td>hot</td>
<td>31</td>
<td>srv_diff_host_rate</td>
</tr>
<tr>
<td>11</td>
<td>num_failed_logins</td>
<td>32</td>
<td>dst_host_count</td>
</tr>
<tr>
<td>12</td>
<td>logged_in</td>
<td>33</td>
<td>dst_host_srv_count</td>
</tr>
<tr>
<td>13</td>
<td>num_compromised</td>
<td>34</td>
<td>dst_host_same_srv_rate</td>
</tr>
<tr>
<td>14</td>
<td>root_shell</td>
<td>35</td>
<td>dst_host_diff_srv_rate</td>
</tr>
<tr>
<td>15</td>
<td>su_attempted</td>
<td>36</td>
<td>dst_host_same_src_port_rate</td>
</tr>
<tr>
<td>16</td>
<td>num_root</td>
<td>37</td>
<td>dst_host_srv_diff_host_rate</td>
</tr>
<tr>
<td>17</td>
<td>num_file_creations</td>
<td>38</td>
<td>dst_host_serror_rate</td>
</tr>
<tr>
<td>18</td>
<td>num_shells</td>
<td>39</td>
<td>dst_host_srv_serror_rate</td>
</tr>
<tr>
<td>19</td>
<td>num_access_files</td>
<td>40</td>
<td>dst_host_rerror_rate</td>
</tr>
<tr>
<td>20</td>
<td>num_outbound_cmds</td>
<td>41</td>
<td>dst_host_rsv_rerror_rate</td>
</tr>
<tr>
<td>21</td>
<td>is_host_login</td>
<td>42</td>
<td></td>
</tr>
</tbody>
</table>
Table 2. Summary of the experiment results for NSL_KDD dataset.

<table>
<thead>
<tr>
<th>Measure</th>
<th>All features</th>
<th>Obtained features</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.8431</td>
<td>0.8815</td>
</tr>
<tr>
<td>error rate</td>
<td>0.1569</td>
<td>0.1185</td>
</tr>
<tr>
<td>fall-out</td>
<td>0.1374</td>
<td>0.0477</td>
</tr>
<tr>
<td>sensitivity/ recall</td>
<td>0.8207</td>
<td>0.8009</td>
</tr>
<tr>
<td>specificity</td>
<td>0.8626</td>
<td>0.9523</td>
</tr>
<tr>
<td>precision</td>
<td>0.8388</td>
<td>0.9365</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.8296</td>
<td>0.8634</td>
</tr>
</tbody>
</table>

Figure 3. Comparison chart of the performance of all features and obtained features for NSL-KDD dataset.

4.2.2 DDoS Dataset

DDoS dataset is used for the evaluation [4]. It contains 27 features, which are labeled as either normal or an attack, as shown in Table 3. The DDoS dataset includes four types of the DDoS attack, which are: Smurf, UDP-Flood, HTTP-Flood and SIDDOS. From this dataset, a small portion of training and testing data is selected for experimentation. The features in bold in Table 3 are the significant features obtained by the feature pruning algorithm used in our experiment on DDoS dataset. Certain features were ignored from the initial feature set due to the same reason mentioned for the NSL-KDD dataset case. Table 4 summarizes the results of the experiment and Figure 4 shows the results in a graphical way.

Table 3. List of features of DDoS dataset.

<table>
<thead>
<tr>
<th>No.</th>
<th>Feature Name</th>
<th>No.</th>
<th>Feature Name</th>
<th>No.</th>
<th>Feature Name</th>
<th>No.</th>
<th>Feature Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>src add</td>
<td>8</td>
<td>flags</td>
<td>15</td>
<td>pkt in</td>
<td>22</td>
<td>utilization</td>
</tr>
<tr>
<td>2</td>
<td>des add</td>
<td>9</td>
<td>fid</td>
<td>16</td>
<td>pktout</td>
<td>23</td>
<td>pkt delay</td>
</tr>
<tr>
<td>3</td>
<td>pkt id</td>
<td>10</td>
<td>seq number</td>
<td>17</td>
<td>pktr</td>
<td>24</td>
<td>pkt send time</td>
</tr>
<tr>
<td>4</td>
<td>from node</td>
<td>11</td>
<td>number of pkt</td>
<td>18</td>
<td>pkt delay node</td>
<td>25</td>
<td>pkt reserved time</td>
</tr>
<tr>
<td>5</td>
<td>to node</td>
<td>12</td>
<td>number of byte</td>
<td>19</td>
<td>pktrate</td>
<td>26</td>
<td>first pkt sent</td>
</tr>
<tr>
<td>6</td>
<td>pkt type</td>
<td>13</td>
<td>node name from</td>
<td>20</td>
<td>byte rate</td>
<td>27</td>
<td>last pkt reserved</td>
</tr>
<tr>
<td>7</td>
<td>pkt size</td>
<td>14</td>
<td>node name to</td>
<td>21</td>
<td>pkt avg size</td>
<td>28</td>
<td></td>
</tr>
</tbody>
</table>

4.2.3 IoTPOT Dataset

IoTPOT dataset is Telnet IoT honeypot that analyzes malware attacks against various IoT devices, such as: IP Camera, DVR, Wireless Router, Customer Premises Equipment, Industrial Video Server, TV
Table 4. Summary of the experiment results for DDoS dataset.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Training/Testing (80/20 %)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All features</td>
</tr>
<tr>
<td>accuracy</td>
<td>0.8985</td>
</tr>
<tr>
<td>error rate</td>
<td>0.1015</td>
</tr>
<tr>
<td>fall-out</td>
<td>0.0098</td>
</tr>
<tr>
<td>sensitivity/recall</td>
<td>0.1071</td>
</tr>
<tr>
<td>specificity</td>
<td>0.9902</td>
</tr>
<tr>
<td>precision</td>
<td>0.5583</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.1797</td>
</tr>
</tbody>
</table>

Figure 4. Comparison chart of the performance of all features and obtained features for DDoS dataset.

Receiver, Heat Pump, Environment Monitoring Unit (EMU system), Digital Video Scalar, home routers [4]. It analyzes the increase in Telnet-based attacks. This dataset contains 9 attack categories and 41 features, as shown in Table 5. The features are extracted by using NetMate tool set. Table 6 summarizes the results of the experiment and Figure 5 shows the results in a graphical way.

Table 5. List of features of IoTPOT dataset.

<table>
<thead>
<tr>
<th>No.</th>
<th>Feature Name</th>
<th>No.</th>
<th>Feature Name</th>
<th>No.</th>
<th>Feature Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>srcip</td>
<td>12</td>
<td>sflow_fbytes</td>
<td>23</td>
<td>std_biat</td>
</tr>
<tr>
<td>2</td>
<td>srcport</td>
<td>13</td>
<td>sflow_bpackets</td>
<td>24</td>
<td>std_active</td>
</tr>
<tr>
<td>3</td>
<td>dstip</td>
<td>14</td>
<td>sflow_bbytes</td>
<td>25</td>
<td>min_fpktl</td>
</tr>
<tr>
<td>4</td>
<td>dstport</td>
<td>15</td>
<td>fpush_cnt</td>
<td>26</td>
<td>mean_fpktl</td>
</tr>
<tr>
<td>5</td>
<td>total_fpackets</td>
<td>16</td>
<td>burg_cnt</td>
<td>29</td>
<td>max_fpktl</td>
</tr>
<tr>
<td>6</td>
<td>total_fvolume</td>
<td>17</td>
<td>bpsh_cnt</td>
<td>28</td>
<td>mean_bpktl</td>
</tr>
<tr>
<td>7</td>
<td>total_bpackets</td>
<td>18</td>
<td>std_bpktl</td>
<td>30</td>
<td>max_bpktl</td>
</tr>
<tr>
<td>8</td>
<td>total_fhlen</td>
<td>19</td>
<td>std_fpktl</td>
<td>31</td>
<td>max_idle</td>
</tr>
<tr>
<td>9</td>
<td>total_bhlen</td>
<td>20</td>
<td>std_bpktl</td>
<td>32</td>
<td>mean_fiat</td>
</tr>
<tr>
<td>10</td>
<td>sflow_fpackets</td>
<td>21</td>
<td>std_fiat</td>
<td>33</td>
<td>max_fiat</td>
</tr>
<tr>
<td>11</td>
<td>std_idle</td>
<td>22</td>
<td>std_idle</td>
<td>34</td>
<td>min_fiat</td>
</tr>
</tbody>
</table>

4.2.4 UNSW_NB15 Dataset

The UNSW-NB 15 dataset was published in 2015 [5]. This dataset contains 9 attack categories and 48 features (shown in Table 7) which are categorized into 6 groups; namely: Flow Features, Basic Features, Content Features, Time Features, Additional Generated Features (General Purpose Features and Connection Features) and Labelled Features. Table 8 summarizes the results of the experiment and Figure 6 shows the results in a graphical way.

Figure 7 shows the ROC curves of the performance of HMM for the obtained features using the following datasets: NSL-KDD, DDoS, IoTPOT and UNSW_NB15. In terms of accuracy and the area
Table 6. Summary of the experiment results for IoTPOI dataset.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Training/Testing (80/20 %)</th>
<th>All features</th>
<th>Obtained features</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.9222</td>
<td>0.9467</td>
<td></td>
</tr>
<tr>
<td>error rate</td>
<td>0.0778</td>
<td>0.0533</td>
<td></td>
</tr>
<tr>
<td>fall-out</td>
<td>0.0069</td>
<td>0.0188</td>
<td></td>
</tr>
<tr>
<td>sensitivity/ recall</td>
<td>0.1312</td>
<td>0.4786</td>
<td></td>
</tr>
<tr>
<td>specificity</td>
<td>0.9931</td>
<td>0.9812</td>
<td></td>
</tr>
<tr>
<td>precision</td>
<td>0.6316</td>
<td>0.6518</td>
<td></td>
</tr>
<tr>
<td>F-measure</td>
<td>0.2173</td>
<td>0.5519</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5. Comparison chart of the performance of all features and obtained features for IoTPOI dataset.

under the ROC curve (AUC), the DDoS dataset achieves the best results. The AUC estimates when the feature pruning method is applied to the datasets are shown in Figure 8.

Table 7. Summary of the experiment results for UNSW_NB15 dataset.

<table>
<thead>
<tr>
<th>No.</th>
<th>Feature Name</th>
<th>No.</th>
<th>Feature Name</th>
<th>No.</th>
<th>Feature Name</th>
<th>No.</th>
<th>Feature Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>srcip</td>
<td>13</td>
<td>dloss</td>
<td>25</td>
<td>trans_depth</td>
<td>37</td>
<td>ct_state_ttl</td>
</tr>
<tr>
<td>2</td>
<td>sport</td>
<td>14</td>
<td>service</td>
<td>26</td>
<td>res_bdy_len</td>
<td>38</td>
<td>ct_flow_http_mthd</td>
</tr>
<tr>
<td>3</td>
<td>dstip</td>
<td>15</td>
<td>sload</td>
<td>27</td>
<td>sjit</td>
<td>39</td>
<td>is_ftp_login</td>
</tr>
<tr>
<td>4</td>
<td>dsport</td>
<td>16</td>
<td>dload</td>
<td>28</td>
<td>djit</td>
<td>40</td>
<td>ct_ftp_cmd</td>
</tr>
<tr>
<td>5</td>
<td>proto</td>
<td>17</td>
<td>spkts</td>
<td>29</td>
<td>stime</td>
<td>41</td>
<td>ct_srv_src</td>
</tr>
<tr>
<td>6</td>
<td>state</td>
<td>18</td>
<td>dpkts</td>
<td>30</td>
<td>stime</td>
<td>42</td>
<td>ct_srv_dst</td>
</tr>
<tr>
<td>7</td>
<td>dur</td>
<td>19</td>
<td>swin</td>
<td>31</td>
<td>sintpkt</td>
<td>43</td>
<td>ct_dst_ltm</td>
</tr>
<tr>
<td>8</td>
<td>sbytes</td>
<td>20</td>
<td>dwin</td>
<td>32</td>
<td>dintpkt</td>
<td>44</td>
<td>ct_svc_ltm</td>
</tr>
<tr>
<td>9</td>
<td>dbytes</td>
<td>21</td>
<td>stcpb</td>
<td>33</td>
<td>tcprrt</td>
<td>45</td>
<td>ct_svc_dport_ltm</td>
</tr>
<tr>
<td>10</td>
<td>sttl</td>
<td>22</td>
<td>dtcpb</td>
<td>34</td>
<td>synack</td>
<td>46</td>
<td>ct_dst_sport_ltm</td>
</tr>
<tr>
<td>11</td>
<td>dtll</td>
<td>23</td>
<td>smean</td>
<td>35</td>
<td>ackdat</td>
<td>47</td>
<td>ct_dst_svc_ltm</td>
</tr>
<tr>
<td>12</td>
<td>sloss</td>
<td>24</td>
<td>dmeans</td>
<td>36</td>
<td>is_sm_ips_ports</td>
<td>48</td>
<td>attack_cat</td>
</tr>
</tbody>
</table>

Table 8. Summary of the experiment results for UNSW_NB15 dataset.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Training/Testing (80/20 %)</th>
<th>All features</th>
<th>Obtained features</th>
</tr>
</thead>
<tbody>
<tr>
<td>accuracy</td>
<td>0.8303</td>
<td>0.9641</td>
<td></td>
</tr>
<tr>
<td>error rate</td>
<td>0.1697</td>
<td>0.0359</td>
<td></td>
</tr>
<tr>
<td>fall-out</td>
<td>0.1916</td>
<td>0.0038</td>
<td></td>
</tr>
<tr>
<td>sensitivity/ recall</td>
<td>0.8459</td>
<td>0.9414</td>
<td></td>
</tr>
<tr>
<td>specificity</td>
<td>0.8084</td>
<td>0.9962</td>
<td></td>
</tr>
<tr>
<td>precision</td>
<td>0.8616</td>
<td>0.9971</td>
<td></td>
</tr>
<tr>
<td>F-measure</td>
<td>0.8536</td>
<td>0.9685</td>
<td></td>
</tr>
</tbody>
</table>
5. CONCLUSION

This paper proposed a feature pruning method for Hidden Markov Models to reduce the number of features and eliminate irrelevant, redundant or noisy features to overcome performance problems and improve the accuracy rate. In addition, this feature pruning method can effectively identify and determine the most significant feature set to be used for classification purposes. Experiment results were tested on four datasets: the NSL_KDD 2009, DDoS 2016, IoTPOT 2016 and UNSW_NB15 2015.
datasets to show the superiority of our approach. The experiments demonstrated that the detection accuracy rate of using our approach is higher than the detection accuracy rate of full feature sets. In addition, false positive rate is lower than in full feature set. NSL_KDD 2009 produces 88.15% accuracy with 10 features. DDoS 2016 achieves 97.41% accuracy with 5 features. IoTpot achieves 94.67% accuracy with 31 features. UNSW_NB15 achieves 96.41% accuracy with 16 features. As for the future work, we will focus on comparing the results derived from this study with other alternative machine learning methods. Meanwhile, we will continue to explore other datasets and flow-based features that could be used in order to improve the performance and achieve higher accuracy.

REFERENCES


ملخص البحث:
إنَّ انتقاء معايير فعالة وذات معنى لنموذج ماركوف المخفي أمر ذو أهمية كبيرة للكشف الشذوذ في قواعد البيانات. يهدف هذا البحث إلى تحديد أبرز المزايا وأهمها في بناء نموذج ماركوف المخفي. ومن أجل تحسين أداء نموذج ماركوف المخفي، يقترح هذا البحث طريقة لتشخيص المزايا. الطريقة المقترحة عالية الدقة تحقق نجاحًا إيجابيًا وفعلاً. ومن ناحية أخرى، فهي قادرة على خفض التعقيد الحسابي وتقليل الوقت دون الإضرار بدقة النموذج. في هذا العمل، تم تطبيق الطرق المقترحة على أربع من مجموعة البيانات (UNSW_NB15; IoTpot; DDoS; NSL-KDD). استخدمها لإجراء دراسة مقارنة تضمن مجموعة البيانات ذات المزايا الكاملة ومجموعة البيانات الفرعية التي تحتوي على المزايا المهمة. وقد أسفرت النتائج التجريبية عن أداء أفضل من حيث الفعالية، وقائمة أعلى، ومعيد أقل لخطأ في الكشف عن المزايا، بالإضافة إلى حذف المزايا المتكررة أو المشوهة.
**Semi-quantitative Security Risk Assessment of Robotic Systems**

Anas AlMajali¹, Khalil M. Ahmad Yousef¹, Bassam J. Mohd¹, Waleed Dweik², Salah Abu Ghalyon¹ and Roa’a Hasan¹

(Received: 28-Aug.-2018, Revised: 15-Oct.-2018 and 5-Nov.-2018, Accepted: 15-Nov.-2018)

**ABSTRACT**

Robots are becoming increasingly integrated in our daily lives, providing services in civilian, industrial and military applications. Many of those applications require robots to be remotely operated and controlled through communication channels. This makes the robotic system susceptible to a class of attacks targeting the connection between the controlling client and the robot, which can render the robot unavailable. The objective of our research is to identify, estimate and prioritize the risks associated with attacks targeting the availability of the robotic system. To achieve our objective, we perform an impact oriented semi-quantitative risk assessment of the loss of availability on the well-known PeopleBot™ mobile robot platform. We experimented with several well-known attacks that can target and affect the availability of the robot. To examine the cyber-physical impacts of the attacks on the robotic system, we setup a ten-goal test area and constructed a 2D map. The robot was programmed to tour the test area while being targeted by cyber-attacks. The physical impacts of the attacks are demonstrated in this paper. The results indicate that attacks can potentially lead to loss of availability which may result in serious cyber-physical consequences.

**KEYWORDS**

Cyber-physical security, Robot, Availability, Threats, Attacks, Vulnerability, Risk, Risk assessment, Mitigation, PeopleBot.

**1. INTRODUCTION**

Robots have been an essential part in many domains like industry, military, research and health [1]. In the Internet of Things (IoT) era, robots are gaining increasing interest to perform critical and non-critical tasks. For example, robots can be used to clean the house floor [2], which is a non-critical task. On the other hand, robots can be used to remotely perform surgical operations, which is considered critical to human life [3]-[4]. As any Cyber-Physical System (CPS), robotic systems are prone to cyber-physical attacks [5]-[6].

Many robotic applications require remote control and monitoring by an operator like Unmanned Aerial Vehicles (UAVs) [7]-[8]. Such applications require establishing a bidirectional communication path between the robot and the controller. The controller can send direct commands to the robot to perform specific functions (e.g. move forward). Based on the application, the controller receives data from the robot (i.e., sensory data), which may be used to make critical decisions, especially in medical and military applications. The communication between the robot and the controller can be wireless (e.g. WiFi - IEEE 802.11) or wired (e.g. Ethernet - IEEE 802.3), utilizing different standards and protocols [9]–[11].

Robotic systems are susceptible to cyber-physical attacks, especially the ones targeting the communication path between the robot and the controller [1]. Attacks on the communication path causing loss of availability are referred to as Denial of Service (DoS) attacks. DoS leads to the loss of the communication between the robot and the controller, as well as the loss of the control and monitoring services. It is important to mention that losing the monitoring and control abilities while performing
critical missions (e.g. surgical operations, defense and space missions) may result in undesirable consequences and harm human lives. This raises the flag and demonstrates the importance of performing risk assessment on robotic platforms, especially if they are responsible for critical missions.

Despite the fact that cyber-physical security is very important, it is usually overlooked. This is evident by the study performed by [12], which indicates that about 30% of the studied robots are accessible from the Internet posing a security threat. The same study also indicates that 76% of the surveyed members did not perform a professional cyber-security assessment on their infrastructure, while more than 50% of the surveyed members did not consider cyber-attacks as a realistic threat. This demonstrates the lack of awareness of the current security posture in the robotic system domain.

The main objective of this paper is to perform cyber-physical security risk assessment on the PeopleBot™ research mobile robot platform [13]. By performing this risk assessment, we raise the awareness of possible threats on the robotic platforms. The robotic system, which is used in this paper as a use-case to perform the risk assessment, consists of the following components (see Figure 1):

- The PeopleBot robot, which runs a server application on its on-board computer to which the client connects. The client then connects to the server (i.e., robot), issues commands and monitors the operation of the robot.
- The client, which is simply a computer machine (e.g. PC or laptop) that runs certain applications to communicate with the robot.
- The network communication medium, which connects the robot and the client. In our case, it is the WiFi access point.
- The attacker, which performs cyber-physical attacks to affect the availability of the robot.

This paper extends our previous work in which we performed qualitative risk assessment of mobile robots [14]. The main contributions of this paper can be summarized as follows:

- We identify possible threats and vulnerabilities that may lead to the loss of availability in the robotic platform using an impact-oriented approach. While those threats and vulnerabilities are not unique to the robotic platform and may apply to any computer network, it is important to study the physical consequences on robotic platforms and the impact on their critical missions.
- We create an experimental test area using the PeopleBot robot. We construct a 2D line and point

![Figure 1. The PeopleBot robotic system under study.](image-url)
map of our research lab environment, identify and localize ten goals within the map for the robot to navigate and tour. We analyze the impact of the attacks on the robot while performing the experimental task of touring the goals. The goals are setup to emulate robotic tasks in a real environment.

- We estimate the risk of each identified threat by following a semi-quantitative approach based on NIST adversarial risk assessment template proposed in [15]. This assessment template was employed in numerous research work, such as the work in [16]–[18]. In this approach, representative numbers are used to estimate the risk of the loss of availability. The main advantage of this approach is that it compromises a middle ground between qualitative and quantitative risk assessments. To the best of our knowledge, this is the first study to perform such a semi-quantitative analysis on vulnerabilities in the robotic platforms.

- We prioritize the risk of each threat based on the risk estimation, so that more severe threats are handled first.

- We discuss the possible physical consequences of the identified threats.

The rest of the paper is organized as follows. In Section 2, related work is discussed. In Section 3, we introduce the PeopleBot mobile robot platform. In Section 4, we present the risk assessment approach used in this paper and the experimental results. Section 5 presents our discussion on the performed security attacks. Finally, we conclude the paper and provide our future directions in Section 6.

2. RELATED WORK

Recently, the security risk assessment on robotic platforms or CPSs has been a very hot research topic. Despite it is a critical issue, the number of publications addressing risk assessment on robotic platforms is limited. In fact, no systemic analysis of industrial robot security was conducted [12]. Further, there is inadequate understanding of what are the actual risks and the affected security goals [19]. In what follows, we present a literature summary for security threat analysis and detection for several robot platforms or CPSs.

Javaid et al. [7] analyzed different security threats to Unmanned Aerial Vehicles (UAVs). The threat model is based on listing vulnerabilities that can affect confidentiality, integrity and availability. The authors followed a typical risk evaluation approach.

Vasconcelos et al. [20] used three DoS attack tools to experimentally evaluate and analyze UAV behavior. The information gathering about the targeted network is performed using a reconnaissance attack that leverages an open source security tool called Network Mapper (Nmap). Next, DoS attacks are launched using Low Orbit Ion Cannon (LOIC), Netwox and Hping3 automated tools. All experiments were conducted in real time on AR.Drone 2.0 UAV while navigating inside a University building. The results show that Hping3 tool causes the highest average network latency of 455.82 milliseconds, which negatively impacts the video streaming application along with other computer vision applications.

Bezzo et al. [21] presented a control-level resiliency estimation technique against security sensor attacks in autonomous robots. The technique is based on a recursive algorithm, which exploits the redundancy in the sensor measurements. Although the proposed approach is generic, the authors used a case study on a vehicle cruise-control, where the latest N velocity measurements are recursively filtered and the variance of the measurements noise is considered to estimate resiliency. In addition, the authors validated the efficacy of their technique through outdoor experiments on two unmanned ground robots.

Bonaci et al. [19] discussed security threats for tele-operated surgical robot Raven II, which is an advanced surgery system. The authors demonstrated that intruders can maliciously control a wide range of the robot functions by performing disruption and manipulation attacks against the surgeon robot communication link that is likely to be wireless. The attacks are based on man-in-the-middle model and they successfully impacted the safety and usability of the surgical robot, which could potentially result in legal and privacy violations. Batson et al. [22] conducted an analysis to identify threats and vulnerabilities in the system’s concept for Unmanned Tactical Autonomous Control and Collaboration (UTACC). Jones and Straub [23] presented a two-stage intrusion detection system (IDS) for detecting
network intrusions and malware in autonomous robots. The authors utilized and trained a deep neural
network to detect commands that deviate from the expected behavior.

Maggi et al. [24] studied the impacts of system-specific attacks on real industrial robots. The authors
analyzed two attacker models: network attacker (i.e., communicates with the robot over the network)
and physical attacker (e.g. the robot operator). Five attack scenarios were demonstrated (e.g. altering
the control-loop parameters, tampering with calibration parameters, …etc.) and the physical impacts in
addition to the compromised requirements (i.e., safety, integrity and accuracy) were discussed.
Similarly, Quarta et al. [12] performed an experimental security analysis of an ABB 6-axis
IRB140/IRC5 industrial robot controller. The authors exploited several software vulnerabilities in the
robot main computer (MC). They mainly exposed and focused on the network services that are essential
for the operation of the robot, such as FTP (File Transfer Protocol).

Lera et al. [25] presented a taxonomy that classifies cyber-security attacks, which target safety and
security of service robots. For safety threats, the proposed taxonomy differentiates between the risks
according to the user type (e.g. domestic, commercial). For each user type, the expected risks are
classified according to the level of the physical impact (e.g. destruction, partial damage) and the origin
of the risk. On the other hand, security threats are classified according to the robot function (i.e., personal
or commercial) and the type of sensors that the robot is equipped with.

Vuong et al. [26] proposed two different approaches for detecting attacks in robotic vehicles. The first
approach was based on using decision trees and the second approach was based on using deep learning.
Loukas et al. [27] argued that the limited rule-based or lightweight machine learning techniques used
for cyber-physical intrusion detection of vehicles can be substituted with more advanced techniques
using computational offloading to the cloud. The boosted processing power is used to implement a deep
multilayer perception and recurrent neural network architecture, which receives the real-time cyber-
physical data captured in the robotic vehicle and analyzes it to detect intrusions. The authors showed
that the deep learning technique noticeably improves the detection accuracy; however, the long detection
latency and the security of the external network between the vehicle and the cloud must be carefully
considered.

Similar to this work, some researchers focused on risk assessment for robotic platforms. Chen et al. [28]
assess the cyber security risks in industrial control systems (ICSs) (e.g. SCADA) by quantifying the
availability using the concept of mean failure cost (MFC). Various security issues arise as the ICS
becomes more integrated with IT networks. Hence, it is important to compare the cost of implementing
security counter-measures with the expected losses of cyber-attacks, especially due to the limited
resources.

Dominic et al. [29] proposed a risk assessment framework for autonomous and cooperative automated
driving. The authors started by describing the recent attack surfaces and then discussing the proposed
application-based threat enumeration and analysis framework. For each threat, model parameters are
specified and accordingly the result vector which characterizes the risk level of the threat is computed.
The result vector reflects attack potential, motivation and impact.

Very recently, the authors in [30] performed qualitative risk assessment of several vulnerabilities
identified specifically to the Adept mobile robots (e.g. the PeopleBot [13]); namely, the
MobileEyes/armiServer client/server robotic applications. Such applications are necessary to establish
the network connection between the Adept robots and their clients or users. In contrast to this work, this
paper proposes a semi-quantitative security risk assessment, where representative numbers are used to
estimate the risk of the loss of availability. The main advantage of this approach is that it compromises
a middle ground between qualitative and quantitative risk assessments. Additionally, the focus of this
paper is mainly on identifying and raising awareness of possible threats and vulnerabilities that may
lead to the loss of availability in the robotic platform under study, the PeopleBot.

3. THE PEOPLEBOT MOBILE ROBOT PLATFORM

Figure 2 shows the PeopleBot mobile robot from Adept company[13], which is used as a case study for
our proposed security risk assessment. The PeopleBot robot is a research platform that can be used in
service and human robot interaction (HRI) projects and in other projects as well [31]. It consists of
multiple hardware and software components: main computer, mechanical actuators, controllers, sensors such as lidars (or laser range finders) and cameras, human-interaction devices, control logic, firmware and operating systems (either Windows 7 or Ubuntu 12.04). The main computer of the PeopleBot is connected to the sensors either through controllers or isolated subnet via an on-board network access point. There are various interfaces or ports on the robot that include-but are not limited to: serial port, Ethernet RJ45 port, USB ports and wireless adapter.

Figure 2. The PeopleBot mobile robot platform, where several sensors are attached to the robot (e.g. pan-tilt-zoom (PTZ) camera, laser range finder, sonars, stereo camera, …etc.).

Several software packages [13] are pre-installed on the robot main computer, such as ARIA, ARNL, MobileEyes, Mapper3, …etc., that enable the control of the robot, its sensors and accessories. In many applications, the robot is often required to be remotely accessible either through a connection to the Internet, or via dedicated wireless access points. Some of those applications include:

- Museum robotic guide application [32].
- Map building and robot self-localization [33]-[34].
- Robotic assistant for healthcare applications [35].
- Gesture-based multi-robot control application and robotic desk clerk application using face recognition [10].
- Application of Myo Armband System to control a robot interface [36].
- Extrinsic calibration of camera and 2D laser sensors without overlap [37].

As it can be seen, all of the above applications of the PeopleBot are very important, critical and thus must be protected against cyber-physical attacks. Consequently, this is one of the main goals of this research.

3.1 Experimental Setup

The experimental setup used to perform the impact-oriented risk assessment on the PeopleBot robotic system (shown in Figure 1) assumes the following components:

- The robot runs a server application on its on-board computer. Such an application is assumed to communicate and access all the sensors pre-installed on the robot, fully control the robot movement, have already established a network connection to a certain network and allow (single or multiple) connections from authorized clients to connect to the robot to access/monitor all of its features, sensors and movement.
- An operator or a client runs certain applications to communicate with the server on his/her own computer.
- The robot network to represent a wireless access point, in which the client can connect to the
server in a client-server mode.

- Only cyber-attacks that can result in the loss of availability on the robot are considered. The attacks target one or more of the following:
  - Application resources; e.g. server application.
  - System resources; e.g. network handling software.
  - Network bandwidth.

To evaluate the impact of the attacks, we created an experimental test area using the PeopleBot robot. We created a 2D line and point map of our research lab environment (6 x 9 m² area). We identified and localized ten goals within the created map of our lab as shown in Figure 3 for the robot to navigate and tour. These goals may emulate important stops for the robot within an industrial workplace application or other applications. The goals are 60 cm spaced apart. The number of goals and separating distances were chosen based on initial feasibility trials to provide adequate accuracy. When a “tour goals” command is issued, the robot tours the goals one by one starting from goal 1. The robot sends its real-time coordinates to the client so that it monitors the physical location of the robot on the map.

![Figure 3. Ten identified and localized goals within the 2D line and point map of our research lab.](image)

4. **RISK ASSESSMENT OF THE PEOPLEBOT ROBOT**

Risk is usually defined as the expected impact of an event on a system or organization [15]. Within the context of this paper, an event refers to a cyber-attack and a system refers to the robotic system. Therefore, the risk determination is a function of the impact of an attack and the probability that this attack occurs and succeeds. The impact of an attack measures the loss of one or more of the main security requirements: confidentiality, integrity and availability. The main focus of this research is only on the availability security requirement. The probability that an attack occurs and succeeds depends on the vulnerabilities and threats of the system under study (i.e., the robot). Hence, the risk determination of a given attack can be expressed as follows:

\[
Risk = Vulnerability \times Threat \times Impact
\]  

Risk assessment is the process of identifying, estimating and prioritizing risks to a system or organization [15]. The flow chart in Figure 4 is adopted from NIST risk assessments [15] and shows the main steps of the assessment process. Next, we present the details of the risk assessment process for the attacks under study.

![Figure 4. Risk assessment process [15].](image)

4.1 **Risk Identification**

To identify the risk, we follow an impact-oriented analysis approach to assess the risk of the attacks on the robot. We focus on the loss of availability as the impact, because this could lead to devastating consequences. As mentioned earlier, the user can connect in a client-server mode to the robot at the
application level to perform certain functions (e.g. transferring medical equipment) and retrieve data (e.g. geographical coordinates). The client and the server have to be available in order to perform the required functions of the robot. Losing availability may have physical impacts, like human injury or physical damage of the robot itself.

4.2 Risk Estimation

To estimate the risk, we have to assess the vulnerabilities and threats that may cause loss of availability. We follow a semi-quantitative risk assessment approach to estimate the risk. In this approach, representative numbers are used to estimate the level of each risk factor, like the severity of a certain vulnerability. The semi-quantitative assessment approach represents a middle-ground between qualitative and quantitative approaches. The semi-quantitative assessment facilitates better comparison and prioritization compared to the qualitative assessment approaches, which rely on non-numerical levels (e.g. low, moderate, high). On the other hand, the semi-quantitative assessment is easier to implement than the quantitative assessment, which requires using specific metrics to measure different risk factors in the assessment process (e.g. the impact of the attack) [15]. Identifying those metrics is challenging in the cyber security domain.

4.3 Risk Prioritization

After estimating the risk level for the attacks under study, the associated threats can be prioritized based on the numerical values calculated to each threat. Next, we present our risk assessment of the loss of availability of the robot.

4.4 Attack Analysis

Figure 5 demonstrates the attack tree for the loss of availability. The root of the attack tree represents the ultimate goal of the attacker, which is causing the loss of availability. In addition, the loss of availability may lead to cyber-physical threats depending on the task that is performed by the robot. The branches of the tree represent the ways and techniques through which the attacker can achieve the ultimate goal. There are three main ways to perform a Denial-of-Service (DoS) attack: attacking the network bandwidth, attacking system resources or attacking the application resources. The next level of the tree represents the techniques that can be used to perform a DoS attack. This attack tree is not inclusive, as it does not cover all possible ways to achieve the loss of availability.

In our assessment, we focus on vulnerabilities that exist in the system itself, especially the network stack. In what follows, we discuss various attacks that target the client, the server or the access point and present the associated experimental results. We want to emphasize that those attacks are not unique to the robotic system. A group of those attacks can target any device that is connected through a network (e.g. SYN flood). On the other hand, another group of those attacks can target devices that are connected through WiFi (e.g. de-authentication attack).

4.4.1 Application Layer Attack

This attack can be implemented on a wireless or wired network. It targets certain application on the server. The attacker sends a large number of requests to the target application on the server, overwhelming its processing and network resources. Typically, HTTP requests are used to exhaust the server. However, the robot does not run a web server, it runs an ARNL server on port 7272 and waits for connections from the client (MobileEyes). An attacker can exploit the ARNL server by initiating a large number of connections at the application level.

Experimental results: we performed a Distributed DoS (DDoS) attack, where seven adversary machines sent application level requests to the server. Initially the robot completed the ten-goal tour. However, after 15 minutes (on average) of the attack, the robot server freezes, old connections are lost and no new connections could be established.

4.4.2 TCP SYN Flood

This attack can be implemented on a wired or wireless network. A TCP SYN flood attack usually targets the operating system’s network handling capability of any server that is listening on a certain port. The
attacker attempts to initiate a large number of TCP connections with the server, where each initiated TCP connection requires what is known as the three-way handshake process (SYN, SYN-ACK, ACK). The server allocates system resources to handle those connections. By overwhelming the server with the attacker’s connections, the server will not have enough resources to handle legitimate connections with the client machine. Hence, the connection with the server can be lost, resulting in the loss of availability of the robot. However, this attack can be easily mitigated by blocking the attacker’s IP address.

On the other hand, the attacker can use IP source address spoofing to improve the attack. In this case, the attacker uses spoofed IP addresses, so that the server responses are sent to devices that are unreachable (i.e., did not initiate the connection). This way, blocking the attacker’s IP does not mitigate the attack. In summary, this kind of attack can result in overwhelming both the operating system of the robot and the robot network.

Experimental results: we performed this attack by sending spoofed TCP SYN flood targeting the robot on port 7272 (ARNL server). Whenever this attack is active, legitimate new connections could not be established from the client to the robot, so no commands could have been issued. Therefore, we could not evaluate the location of the robot as the command could not be issued in the first place. If the connection had already been established before the attack, then the attack fails.

4.4.3 IPv6 Router Advertisement Flooding

In IPv6 Router Advertisement (RA) floods, the adversary floods the robot network with fake IPv6 RA packets or messages. Any computer, including the robot, running a vulnerable operating system (on which IPv6 routing is enabled), will be overwhelmed with those fake IPv6 RA packets. In order to perform this attack, the adversary should be connected to the same network of the victim (i.e., the robot network).

Experimental results: the results of this attack varied based on the vulnerabilities of the underlining operating system of the robot. We performed an IPv6 attack while running Windows 7 on the robot. Windows 7 is vulnerable to this attack. First, we issue a "tour goals" command from the client machine and immediately start the attack. Subsequently, after starting the attack, the robot operating system freezes, the robot stops before reaching goal 2 and the client loses connection with the robot. To restart the robot, it requires hard reset. Ubuntu operating system is not vulnerable to this type of attack. If client machine is also running a vulnerable operating system, then it will also freeze.

4.4.4 ICMP Echo Request

This attack works in the network layer. The attacker floods the robot or the client with ICMP echo request (ping) packets overwhelming the network of the robot. This degrades the network performance and causes legitimate traffic to be lost rendering the robot unavailable. To improve the attack, the attacker can use IP source address spoofing.
Experimental results: we performed a DDoS attack, where seven adversary machines sent spoofed ICMP echo requests to the server. Initially, the robot completed the ten goal tour. However, after 45 minutes (on average) of the attack, the robot server freezes, old connections are lost and no new connections could be established.

4.4.5 De-authentication Attack

In a de-authentication attack, the attacker sends a de-authentication frame to the victim machine with a spoofed source address of the access point to terminate the victim’s connection with the access point. This results in losing the WiFi connection and disconnecting the client from the server. This attack does not require encryption, so it can be performed by an outsider who can sniff the WiFi connection to get the MAC address of the victims and the access point [38].

Experimental results: we performed this attack after issuing a "tour goals" command from the client. The robot and client were immediately disconnected from the network. The client's map showed that the robot did not reach goal 2. However, this attack did not stop the robot from physically touring all the goals.

4.4.6 Jamming Attack

If the robot and client are connected using WiFi (IEEE 802.11), then they are susceptible to jamming attacks. This kind of attack requires that the attacker be in close proximity to the network under attack. The jammer has to operate in the 2.4 GHz or 5 GHz radio frequency which is used by the WiFi technology. This attack was not performed in our research, as it is a pure physical layer test which is beyond the scope of this paper.

4.4.7 Attack Summary

We specifically chose the previous attacks, because they demonstrate various ways to cause loss of availability (DoS). With the exception of the jamming attack, no special equipment is necessary and script kiddies can implement those attacks (e.g. using Kali Linux [39]). Table 1 summarizes attack characteristics and results. It demonstrates the physical domain of the attack (wired/wireless) and the targeted resources by the attack. Furthermore, Table 1 lists the impact of the attacks on the robot.

4.5 Semi-quantitative Risk Assessment Determination

As mentioned earlier in this section, it is convenient to use a semi-quantitative risk assessment approach. Table 2 represents the mapping between the qualitative and semi-quantitative values [15]. Following [15], we choose to use the semi-quantitative values in the range between 0 and 10, where 10 represents the maximum severity of vulnerability or impact.

The overall likelihood of any threat is assigned a value between 0 and 1 based on the likelihood of attack initiation and the likelihood of success of the initiated attack. Consequently, the risk level is the product of the overall likelihood and the impact of the attack, as demonstrated in Equation 2.

\[
Risk = \text{Overall Likelihood} \times \text{Impact};
\]

where the overall likelihood of a threat is computed as a function of the likelihood of attack initiation and the likelihood of success of the initiated attack.

Now, we discuss how to generate the risk table, which summarizes the risk assessment process. We used the adversarial risk assessment template proposed in [15]. Table 3 presents the end result of the risk assessment process for the threat events under study. Following, each item in Table 3 is discussed (for more details about those items please refer to tables D-3, D-4, D-5, E-4, F-2, F-5 and H-3 in [15]):

**Threat Event**: refers to the threat that is currently being analyzed.

1) **Threat Sources**: refer to the threat source, which can be an insider, outsider, trusted insider or privileged insider [15]. Some attacks require the attacker to communicate with the robot, client or access point. This implies that the attacker has to be on the same network of those devices (unless the robot is configured to be remotely accessed, like having a public IP address). This applies to application level attacks, TCP SYN floods, IPv6 RA floods and ICMP echo requests.
Table 1. Summary of threat characteristics and experimental results

<table>
<thead>
<tr>
<th>Threat</th>
<th>Wireless or wired</th>
<th>Target resources</th>
<th>Impact on the robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application level attack</td>
<td>Wireless</td>
<td>Application resources</td>
<td>Connection with the robot is lost. No new connections could be established.</td>
</tr>
<tr>
<td>TCP SYN flood attack</td>
<td>Both</td>
<td>System resources and/or network bandwidth</td>
<td>No new legitimate client connections could be established with the robot.</td>
</tr>
<tr>
<td>IPv6 RA attack</td>
<td>Both</td>
<td>System resources</td>
<td>OS (Widows 7) freezes, robot stops before goal 2 and robot disconnected from client.</td>
</tr>
<tr>
<td>ICMP echo request flood</td>
<td>Both</td>
<td>Network bandwidth</td>
<td>Connection with the robot is lost. No new connections could be established.</td>
</tr>
<tr>
<td>De-authentication attack</td>
<td>Wireless</td>
<td>System resources</td>
<td>Robot disconnected from client before reaching goal 2 and robot is moving without supervision.</td>
</tr>
<tr>
<td>Jamming attack</td>
<td>Wireless</td>
<td>Network bandwidth</td>
<td>No connection between the client and the robot.</td>
</tr>
</tbody>
</table>

Table 2. Mapping of the qualitative levels to the semi-quantitative levels

<table>
<thead>
<tr>
<th>Qualitative level</th>
<th>Very High</th>
<th>High</th>
<th>Moderate</th>
<th>Low</th>
<th>Very Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-quantitative level</td>
<td>10</td>
<td>8</td>
<td>5</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

On the other hand, de-authentication and jamming attacks can be conducted by outsiders with close proximity to the target machines.

2) Capability: this is one of the characteristics of the threat source. An attacker with high capability is one with high level of expertise and is well-resourced. Because of the increasing interest in cyber-security, there is an increasing number of highly capable threat sources, especially in a scientific and research environment. The reader should note that the attacks we are implementing can be implemented by a person with moderate or low capabilities.

3) Intent: this is one of the characteristics of the threat source. The adversary seeks to undermine critical functions of the system and may result in physical damage by causing loss of availability.

4) Targeting: this is one of the characteristics of the threat source. The adversary targets a specific mission or function within an organization (i.e., the target of the attack is not random; however, the threats discussed in this work are not unique to the robots).

5) Relevance: indicates how relevant the threat event is to the system under study. For example, if the threat event already happened in the system, then it is confirmed. As we do not have evidence that those threats happened (i.e., reported in the literature as real attack) on a PeopleBot robot, we rank them as possible.

6) Likelihood of attack initiation: refers to the likelihood that the adversary will initiate the attack. Hence, it depends on adversary intend, capability and targeting [15]. To find this likelihood, we compute the normalized average of those three values as shown in Equation 3.

\[
\text{normalized average} = \frac{\text{computed average}}{10}
\]

7) Severity of vulnerabilities: all of our implemented attacks make use of the vulnerabilities in the communication link with the robot, which are exploitable and exposed. Application level, TCP SYN floods and ICMP echo request can be mitigated by blocking the source IP address of the attacker. However, if IP address spoofing is used, then protecting against those attacks will be difficult. As such, the severity of the attacks is ranked high. IPv6 RA floods can be easily mitigated by disabling IPv6 routing, so it is ranked as moderate. In addition, Windows operating
systems are highly affected by this attack, while Linux operating systems are moderately affected, because they only take the first 15 route advertisements and ignore the remaining. Deauthentication and jamming attacks are very difficult to be stopped [40], [41].

8) **Pervasiveness of predisposing conditions:** vulnerabilities that apply to all robots running the same setup are ranked high.

9) **Likelihood Initiated Attack Succeeds:** This item depends on the threat source capability, severity of the vulnerability and predisposing conditions. To compute the probability of a successful initiated attack, we average those three factors and then normalize them.

10) **Overall Likelihood:** this is a combination of the likelihood of attack initiation and the likelihood of the successful initiated attack. In order to capture the effect of both likelihoods, we average the two values.

11) **Level of impact on the robot:** severe or catastrophic effect on the robot system means high impact. All the attacks discussed in this paper cause high impact, which is the loss of availability of the robot.

12) **Risk:** this is the final risk assessment measure, which is the product of the overall likelihood and the level of impact (Equation 2). The risk value represents an estimation of the risk of each threat and can be used to prioritize risk handling; for example, (application level attack):

   - The threat source characteristics are all ranked high and hence they were assigned a value of 8 based on Table 2.
   - The likelihood of attack initiation is computed based on Equation 3 using the threat source characteristics values (i.e., $\frac{8+8+8}{10} = 0.8$).
   - The severity of vulnerabilities and the pervasiveness of predisposing conditions were all ranked high and hence they are assigned a value of 8.
   - The likelihood that the initiated attack succeeds is computed based on Equation 3 using the threat source capability, severity of vulnerabilities and the pervasiveness of predisposing conditions values (i.e., $\frac{8+8+8}{10} = 0.8$).
   - The overall likelihood is computed by taking the average of the likelihood of attack initiations and the likelihood that the initiated attack succeeds (i.e., $\frac{0.8+0.8}{2} = 0.8$).
   - The level of impact on the robot is ranked high and thus is assigned a value of 8.
   - The risk level is computed based on Equation 2 (i.e., $0.8 \times 8 = 6.4$).

In the following section, we discuss the physical consequences for the loss of availability on the robot.

### 5. Discussion

In this section, we discuss the physical consequences of the loss of availability on the robot. Since this is an application-specific process, we consider several applications of the robot. We consider the loss of availability as the impact of the attack in the risk assessment process. The impact values in the risk assessment process can be assigned based on the specific robotic application. For example, if the robot freezes in a high traffic area (e.g. airport application), then this may cause more damage than if it continues with the current task. On the other hand, it may be safer for the robot to stop and abort its current task if moving with precise supervision is required (e.g. industrial applications).

#### 5.1 Hospital Application

This application is for the case when the robot is operated in a hospital that is fully covered by a wireless LAN. In this application, the robot is assumed to be used in two critical missions: creating a 2D map of the hospital and performing time-sensitive tasks (e.g. delivering medical supplies and equipment) [42]. Both missions require the robot to be remotely controlled and monitored by a human operator through the wireless LAN. Creating a map requires logging lots of data generated from the 2D laser range finder sensor and the wheel encoders of the robot [33]. The operator connects to the robot through an SSH
Table 3. Risk Assessment Table of the PeopleBot robot

<table>
<thead>
<tr>
<th>Threat Event</th>
<th>Threat Source Characteristics</th>
<th>Capability</th>
<th>Intent</th>
<th>Targeting</th>
<th>Relevance</th>
<th>Likelihood of attack initiation</th>
<th>Severity of vulnerabilities</th>
<th>Pervasiveness of predisposing conditions</th>
<th>Likelihood of Initiated Attack</th>
<th>Likelihood of Overall Success</th>
<th>Overall Level of Impact on robot</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application level attack</td>
<td>Insider</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>Possible</td>
<td>0.80</td>
<td>8</td>
<td>8</td>
<td>0.80</td>
<td>0.80</td>
<td>8</td>
<td>6.40</td>
</tr>
<tr>
<td>TCP SYN flood attack</td>
<td>Insider</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>Possible</td>
<td>0.80</td>
<td>8</td>
<td>8</td>
<td>0.80</td>
<td>0.80</td>
<td>8</td>
<td>6.40</td>
</tr>
<tr>
<td>IPv6 RA floods</td>
<td>Insider</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>Possible</td>
<td>0.80</td>
<td>2</td>
<td>8</td>
<td>0.60</td>
<td>0.70</td>
<td>8</td>
<td>5.60</td>
</tr>
<tr>
<td>ICMP echo request attack</td>
<td>Insider</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>Possible</td>
<td>0.80</td>
<td>8</td>
<td>8</td>
<td>0.80</td>
<td>0.80</td>
<td>8</td>
<td>6.40</td>
</tr>
<tr>
<td>De-authentication attack</td>
<td>Insider or outsider</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>Possible</td>
<td>0.80</td>
<td>10</td>
<td>8</td>
<td>0.87</td>
<td>0.84</td>
<td>8</td>
<td>6.72</td>
</tr>
<tr>
<td>Jamming attack</td>
<td>Insider or outsider</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>Possible</td>
<td>0.80</td>
<td>10</td>
<td>8</td>
<td>0.87</td>
<td>0.84</td>
<td>8</td>
<td>6.72</td>
</tr>
</tbody>
</table>

session and runs a data logging program, such as “sickLogger”[43]. Moreover, the operator runs a monitoring program to control the robot movement using image sequences from the camera sensor. While running these applications, multiple physical consequences are possible if the robot becomes unavailable as a result of exploiting the attacks discussed in Subsection 4.4. First, the robot may get physically damaged due to open stairs in the hospital environment. Second, sudden accidents may cause human injuries or damage critical equipment.

On the other hand, performing time-sensitive delivery tasks requires the operator and the robot to connect using client-server programs; “MobileEyes” and “arnlServer”[43]. The robot runs the ARNLServer program which loads the hospital map in the robot’s memory then waits for commands from the client program. The client runs the MobileEyes program, which connects to the ARNLServer and navigates the robot to destinations within the map using IR, bumpers, camera, sonars and laser range finder sensors. One critical physical consequence of loss of availability in this scenario is delaying or preventing the robot delivering urgent medical supplies to the hospital operation room. Another critical consequence is that the robot may get physically hijacked.

5.2 Airport Application

This application considers the case when the robot is operated at airports to perform two critical tasks: delivering passengers’ luggage to their respective terminals [44] and carrying out security checks for travelers [45]. To achieve the first task, the robot needs to create an airport map for luggage delivery. Creating the airport map can be done exactly the same fashion in the hospital scenario; hence, similar physical consequences might occur after a successful attack that leads to loss of availability. Once the map is constructed, the “MobileEyes” and “ARNLServer” client-server applications are used to load the map in the robot’s memory and send navigation commands to the robot for delivery purposes. Losing availability while delivering passengers’ luggage might result in luggage being delayed, which causes customer dissatisfaction and impacts the airline public image. More severely, the luggage might maliciously get stolen, replaced or delivered to incorrect terminals.
To achieve the second task (i.e., performing security checks for travelers), the robot uses its high-definition cameras and facial recognition software probably running on the operator’s computer. Losing availability while carrying out such critical task has the dire consequence of allowing suspects to escape the security check.

5.3 Industrial Application

This application considers the case when the robot is operated at an industrial warehouse for helping in the production of expensive instruments and equipment. This is one of the critical scenarios, as the robot needs to make precise movements and actions. Losing availability of the robot could result in a financial impact. In this case, financial losses can be huge due to production halt, business disruption and replacement or remediation costs. It is clear that the physical consequences described above can lead to catastrophic results. Security experts need to carefully consider taking well-studied countermeasures to reduce or even prevent attacks. Surely this needs to be considered per each application of the robotic platform.

5.4 Mitigating Robot Cyber-Security

In this subsection, we suggest some recommendations as possible mitigation techniques for the loss of availability of the robot. It is important to mention that building a secure robot is not a simple task. However, considering and implementing the following recommendations can highly improve the cyber-security of the robot.

- Encrypt the robot communications.
- Assure that only authorized users have access to the robot network, on-board computer, services and functionality.
- Install operating systems updates to fix known vulnerabilities.
- Invest in cyber-security education for everyone using the robot.
- Enforce a backup plan policy in case the robot becomes unavailable.
- Enable SYN Cookies to protect against TCP SYN floods [46].

5.5 Limitations

The main limitation of this approach is the dependency on identifying the vulnerabilities of the system. This requires deep knowledge of all the systems integrated to operate the robot, including hardware, software, operating systems, communications, …etc.

6. CONCLUSIONS

In this paper, we identified, estimated and prioritized the risks associated with attacks targeting the availability of the robotic system.

The paper discussed several attacks that can result in losing the availability of the PeobleBot robot. We setup a ten-goal test area in our research lab. The robot is commanded to tour the ten goals while being attacked. We discussed the experimental results from each attack. The paper presented an impact-oriented analysis approach to assess the risk of these attacks as demonstrated in Table 3. We discussed the physical impacts for losing the availability of the robot while performing several critical applications. The severity of the physical impacts raises the flag of the requirement of considering effective cyber-security countermeasures.

Future work could focus on extending the semi-quantitative risk assessment to a quantitative assessment. In addition, one area of research is to examine attacks on integrity and confidentiality of robotic systems. We also plan to analyze the resilience of robotic systems to cyber-physical attacks. By analyzing the resilience, we investigate how the robotic system responds and recovers from failures that are caused by cyber-physical attacks.

ACKNOWLEDGEMENTS

This research was supported by a grant from the Deanship of Scientific Research at the Hashemite University, Zarqa, Jordan.
REFERENCES


ملخص البحث:
أستصبحت الروبوتات تتكاملك بشكل متزايد في حياتنا اليومية، حيث تقدم الخدمات في التطبيقات المدنية والصناعية والعسكرية. العديد من هذه التطبيقات تتطلب أن يتم تشغيل الروبوتات والتحكم فيها عن بعد من خلال قنوات الاتصال. وهذا يجعل نظام الروبوت عرضة لطبيعة من الهجمات التي تهدف إلى الاتصال بين العميل المسيطر والروبوت، مما يجعل الروبوت غير متوفر. الهدف من هذه الدراسة هو تحديد وتقدير المخاطر المرتبطة بالهجمات التي تهدف توفر الروبوت من أجل تحقيق هدفنا، فمنا بإجراء تقييم مخاطر شبه كمي موجه نحو تأثير فقدان توفر الروبوت (PeopleBot™). فمنا بإجراء مجموعة من التجارب التي استخدمنا فيها العديد من الهجمات المعروفة التي يمكن أن تستهدف وتؤثر على توفر الروبوت. وقد أثبتت النتائج أن الهجمات يمكن أن تؤدي إلى فقدان توفر الروبوت، الأمر الذي يؤدي بدوره إلى عواقب خطيرة.
<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abdelhak Lakhouaja</td>
<td>Mohammed First University</td>
<td>Morocco</td>
</tr>
<tr>
<td>Abdellah Youssfi</td>
<td>Mohamed V University</td>
<td>Morocco</td>
</tr>
<tr>
<td>Abdul Malik Al-Salam</td>
<td>King Saud University</td>
<td>Saudi Arabia</td>
</tr>
<tr>
<td>Abdul Rahim Ahmad</td>
<td>UNITEC, Malaysia</td>
<td>Malaysia</td>
</tr>
<tr>
<td>Ahmad T. Al-Taani</td>
<td>Yarmouk University, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Ahamed Rafaa AUC, Egypt</td>
<td></td>
<td>Italy</td>
</tr>
<tr>
<td>Alessandro Rizzi</td>
<td>University of Milan</td>
<td>Italy</td>
</tr>
<tr>
<td>Ali Younis Maqousi</td>
<td>University of Petra, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Anqi Cui</td>
<td>RSVP Technologies Inc., Canada</td>
<td>Canada</td>
</tr>
<tr>
<td>Antonio Navarro</td>
<td>Aveiro University, Portugal</td>
<td>Portugal</td>
</tr>
<tr>
<td>Artur Babiarz</td>
<td>Silesian University of Technology, Poland</td>
<td>Poland</td>
</tr>
<tr>
<td>Benjamin Aziz</td>
<td>University of Portsmouth, UK</td>
<td>UK</td>
</tr>
<tr>
<td>Cheng Li</td>
<td>William &amp; Mary, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Chunnie Lai</td>
<td>Howard University, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Danilo Mandi</td>
<td>Imperial College London, UK</td>
<td>UK</td>
</tr>
<tr>
<td>Degan Zhang</td>
<td>TU TE, China</td>
<td>China</td>
</tr>
<tr>
<td>Delina Beh Mei</td>
<td>UniKL, Malaysia</td>
<td>Malaysia</td>
</tr>
<tr>
<td>Devisas Pardhan</td>
<td>Acharaya Institutes, India</td>
<td>India</td>
</tr>
<tr>
<td>Driss Namly</td>
<td>Mohammed V University, Morocco</td>
<td>Morocco</td>
</tr>
<tr>
<td>Duangjai Jitkongchuen</td>
<td>Dhurakij Pundit University, Thailand</td>
<td>Thailand</td>
</tr>
<tr>
<td>Duygu Sinanc Terzi</td>
<td>Gazi University, Turkey</td>
<td>Turkey</td>
</tr>
<tr>
<td>E. Uzun</td>
<td>PBB ETU, Turkey</td>
<td>Turkey</td>
</tr>
<tr>
<td>El Houssine Chouaib</td>
<td>Hanik, NIBIO, Norway</td>
<td>Norway</td>
</tr>
<tr>
<td>Emad M. Al-Shawafaka</td>
<td>Yarmouk University, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Evandro Cunha</td>
<td>UFMG, Brazil</td>
<td>Brazil</td>
</tr>
<tr>
<td>Fadoua Khennou</td>
<td>Sidi Mohamed Ben Abdellah University, Morocco</td>
<td>Morocco</td>
</tr>
<tr>
<td>Mehmet Fatih CAGLAR</td>
<td>Sileyman Demirel University, Turkey</td>
<td>Turkey</td>
</tr>
<tr>
<td>Fawaz S.Al-Anzi</td>
<td>Kuwait University, Kuwait</td>
<td>Kuwait</td>
</tr>
<tr>
<td>Firas A. Jassim</td>
<td>Harvard University, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Floriano De Rango</td>
<td>University of Calabria, Italy</td>
<td>Italy</td>
</tr>
<tr>
<td>Frans Hendrik Botes</td>
<td>CPUT, South Africa</td>
<td>South Africa</td>
</tr>
<tr>
<td>George Pallis</td>
<td>University of Cyprus, Cyprus</td>
<td>Cyprus</td>
</tr>
<tr>
<td>Ghasan G. Kanaan</td>
<td>Amman Arab University, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Gudvada Viswanath Raviteja</td>
<td>GITAM University, India</td>
<td>India</td>
</tr>
<tr>
<td>Gunasekaran Manogaran</td>
<td>University of California, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Hamid Reza Hassani</td>
<td>Shahed University, Iran</td>
<td>Iran</td>
</tr>
<tr>
<td>Hani Abu-Salem</td>
<td>University of South Carolina, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Heekuck Oh</td>
<td>Hanyang University, Korea</td>
<td>Korea</td>
</tr>
<tr>
<td>Hosein M. Golshan Mojdehi</td>
<td>University of Denver, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Hossam M. Zawbaa</td>
<td>Beni-Suef University, Egypt</td>
<td>Egypt</td>
</tr>
<tr>
<td>Ibhera Medeiros</td>
<td>University of Lisbon, Portugal</td>
<td>Portugal</td>
</tr>
<tr>
<td>Imad Zeroual</td>
<td>Mohamed First University, Morocco</td>
<td>Morocco</td>
</tr>
<tr>
<td>Ines Zribi</td>
<td>University of Sfax, Tunisia</td>
<td>Tunisia</td>
</tr>
<tr>
<td>Ioannis Agrafiotis</td>
<td>University of Oxford, UK</td>
<td>UK</td>
</tr>
<tr>
<td>Irini Fundulaki</td>
<td>Foundation for Research and Technology, Greece</td>
<td>Greece</td>
</tr>
<tr>
<td>Iztok Jr Fister</td>
<td>University of Maribor, Slovenia</td>
<td>Slovenia</td>
</tr>
<tr>
<td>Jamal Nahar Bani Salameh</td>
<td>Matlah University, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Jawad K. Ali</td>
<td>University of Technology, Iraq</td>
<td>Iraq</td>
</tr>
<tr>
<td>Jin-Fu Li</td>
<td>National Central University, Taiwan</td>
<td>Taiwan</td>
</tr>
<tr>
<td>Jorge Dias</td>
<td>Khalifa University, UAE</td>
<td>UAE</td>
</tr>
<tr>
<td>K. Anitha Kumari</td>
<td>PSG College of Technology, Egypt</td>
<td>Egypt</td>
</tr>
<tr>
<td>Kalman Graffi</td>
<td>HHUD, Germany</td>
<td>Germany</td>
</tr>
<tr>
<td>Ke Gu, BJUT, China</td>
<td></td>
<td>China</td>
</tr>
<tr>
<td>Lilly Suriani Affendey</td>
<td>UPM, Malaysia</td>
<td>Malaysia</td>
</tr>
<tr>
<td>Luanna Lopes Lobato</td>
<td>Federal University of Goias, Brazil</td>
<td>Brazil</td>
</tr>
<tr>
<td>M. B. R. Murthy</td>
<td>RGUKT, India</td>
<td>India</td>
</tr>
<tr>
<td>Maha Alrabiah, Al Imam Mohammad ibn Saud Islamic University, Saudi Arabia</td>
<td>University of Technology,</td>
<td></td>
</tr>
<tr>
<td>Mahmoud Al-Ayyoub</td>
<td>JUST, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Mahmoud El-Haj</td>
<td>Lancaster University, UK</td>
<td>UK</td>
</tr>
<tr>
<td>Majed Said Al-Safiya</td>
<td>AAU, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Manisha Malhotra</td>
<td>Chandigarh University, India</td>
<td>India</td>
</tr>
<tr>
<td>Massimo Tosa</td>
<td>Technical University of Kaiserslautern, Germany</td>
<td>Germany</td>
</tr>
<tr>
<td>Meenupriya Swaminathan</td>
<td>Northeastern University, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Mehmet A. Belen</td>
<td>Yildiz Technical University, Turkey</td>
<td>Turkey</td>
</tr>
<tr>
<td>Michel R. V. Chaudron</td>
<td>University of Gothenburg, Sweden</td>
<td>Sweden</td>
</tr>
<tr>
<td>Minxian Xu</td>
<td>University of Melbourne, Australia</td>
<td>Australia</td>
</tr>
<tr>
<td>Mohammad Tarigul Islam</td>
<td>UKM, Malaysia</td>
<td>Malaysia</td>
</tr>
<tr>
<td>Mohamed Mahmood</td>
<td>HICIT, Shorouk Academy, Egypt</td>
<td>Egypt</td>
</tr>
<tr>
<td>Mohammad H. Alomari</td>
<td>ASU, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Mohammad Najj Al-Kabi</td>
<td>Zarga University, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Mousa I. Hussein</td>
<td>UEA University, UAE</td>
<td>UAE</td>
</tr>
<tr>
<td>Nada Ghneim</td>
<td>Syrian Virtual University, Syria</td>
<td>Syria</td>
</tr>
<tr>
<td>Najmiah Radiaah Mohamad</td>
<td>UTEM, UEA</td>
<td>Malaysia</td>
</tr>
<tr>
<td>Narinder Singh</td>
<td>Punjab University, India</td>
<td>India</td>
</tr>
<tr>
<td>Nizar Habash</td>
<td>New York University, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Osama M.F. Abu-Shakhk</td>
<td>PSUT, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Osama Mohamed Elrabi</td>
<td>Misrata University, Libya</td>
<td>Libya</td>
</tr>
<tr>
<td>P. S. Ashitankar</td>
<td>KIITS, India</td>
<td>India</td>
</tr>
<tr>
<td>Paolo Rosso</td>
<td>Technical University of Valencia, Spain</td>
<td>Spain</td>
</tr>
<tr>
<td>Pei Liao</td>
<td>Northeastern University, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Peipei Wang</td>
<td>NC SU, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Petra Leimich</td>
<td>Edinburgh Napier University, UK</td>
<td>UK</td>
</tr>
<tr>
<td>Prayoot Akkaeraekthalin</td>
<td>KMUTNB, Thailand</td>
<td>Thailand</td>
</tr>
<tr>
<td>Priti Sehgal</td>
<td>University of Delhi, India</td>
<td>India</td>
</tr>
<tr>
<td>Qannmer H. Abbasi</td>
<td>University of Glasgow, UK</td>
<td>UK</td>
</tr>
<tr>
<td>Qi Han, HIT, China</td>
<td></td>
<td>China</td>
</tr>
<tr>
<td>Qi Jiang</td>
<td>Xiand University, China</td>
<td>China</td>
</tr>
<tr>
<td>Quan Leng</td>
<td>University of Texas, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Rabii Ayed</td>
<td>ISGS, Tunisia</td>
<td>Tunisia</td>
</tr>
<tr>
<td>Radio-Emil Precup</td>
<td>Polytechnic of Timisoara, Romania</td>
<td>Romania</td>
</tr>
<tr>
<td>Razali Ngah, UTM, Malaysia</td>
<td></td>
<td>Malaysia</td>
</tr>
<tr>
<td>Riad Sonbol</td>
<td>HISt, Syria</td>
<td>Syria</td>
</tr>
<tr>
<td>Richard McClatchey</td>
<td>University of the West of England, UK</td>
<td>UK</td>
</tr>
<tr>
<td>Rodrigo Roman</td>
<td>University of Malaga, Spain</td>
<td>Spain</td>
</tr>
<tr>
<td>Rohit Salgotra</td>
<td>ITET, India</td>
<td>India</td>
</tr>
<tr>
<td>Roshan Ragel</td>
<td>University of Peradeniya, Sri Lanka</td>
<td>Sri Lanka</td>
</tr>
<tr>
<td>S. K. Bharti</td>
<td>University of Delhi, India</td>
<td>India</td>
</tr>
<tr>
<td>Saad Mustafa</td>
<td>COMSATS Institute of Information Technology, Pakistan</td>
<td>Pakistan</td>
</tr>
<tr>
<td>Sajjad Mohammad-Ali-Nezhad</td>
<td>University of Qom, Iran</td>
<td>Iran</td>
</tr>
<tr>
<td>Salvatore Sessa</td>
<td>University of Naples Federico II, Italy</td>
<td>Italy</td>
</tr>
<tr>
<td>Sandeep Kumar Samal</td>
<td>Georgia Institute of Technology, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Sankalap Arora</td>
<td>DAV University, India</td>
<td>India</td>
</tr>
<tr>
<td>Sarawuth Chaimool</td>
<td>KMUTNB, Thailand</td>
<td>Thailand</td>
</tr>
<tr>
<td>Saru Kumari</td>
<td>CCS University, India</td>
<td>India</td>
</tr>
<tr>
<td>Sayed Af Arif Ali</td>
<td>Mewar University, India</td>
<td>India</td>
</tr>
<tr>
<td>Shahrul Azman</td>
<td>UKM, Malaysia</td>
<td>Malaysia</td>
</tr>
<tr>
<td>Shalesh D. Kamble</td>
<td>YCCE, India</td>
<td>India</td>
</tr>
<tr>
<td>Shubham Gupta</td>
<td>Facebook, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Simona Bernardi</td>
<td>University of Zaragoza, Spain</td>
<td>Spain</td>
</tr>
<tr>
<td>Syed Ahmed Raza</td>
<td>Western University, Canada</td>
<td>Canada</td>
</tr>
<tr>
<td>Thomas De Cruulde</td>
<td>KU Leuven, Belgium</td>
<td>Belgium</td>
</tr>
<tr>
<td>Thomas Pasquier</td>
<td>University of Cambridge, UK</td>
<td>UK</td>
</tr>
<tr>
<td>Tolga Ensari</td>
<td>Istanbul University, Turkey</td>
<td>Turkey</td>
</tr>
<tr>
<td>Vladimir B. Kovacevic</td>
<td>University of Belgrade, Serbia</td>
<td>Serbia</td>
</tr>
<tr>
<td>Weiqing Sun</td>
<td>University of Toledo, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Xavier Luciani</td>
<td>University of Toulon, France</td>
<td>France</td>
</tr>
<tr>
<td>Xuechao Ming</td>
<td>TU TE, China</td>
<td>China</td>
</tr>
<tr>
<td>Yaser Al-Lahham</td>
<td>Zaraqa University, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Yaser Jararweh</td>
<td>JUST, Jordan</td>
<td>Jordan</td>
</tr>
<tr>
<td>Yin Minn Pu Bo’s</td>
<td>Yokohama National University, Japan</td>
<td>Japan</td>
</tr>
<tr>
<td>Yu Zhanga</td>
<td>Stanford University, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Yuntao Wu</td>
<td>WIT, China</td>
<td>China</td>
</tr>
<tr>
<td>Zeashan Hameed Khan</td>
<td>Bahria University, Pakistan</td>
<td>Pakistan</td>
</tr>
<tr>
<td>Zhaszeng Cui</td>
<td>ETH Zurich, Switzerland</td>
<td>Switzerland</td>
</tr>
<tr>
<td>Zhaoqiang Xia</td>
<td>NPU, China</td>
<td>China</td>
</tr>
<tr>
<td>Zhen Chen</td>
<td>University of Rhode Island, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Zijuang Hao</td>
<td>William &amp; Mary, USA</td>
<td>USA</td>
</tr>
<tr>
<td>Zoltan Adam Mann</td>
<td>Palumbo, Germany</td>
<td>Germany</td>
</tr>
</tbody>
</table>