In the Name of God

Journal of
Information Systems & Telecommunication
Vol. 8, No.1, January-March 2020, Serial Number 29

Research Institute for Information and Communication Technology
Iranian Association of Information and Communication Technology
Affiliated to: Academic Center for Education, Culture and Research (ACECR)

Print ISSN: 2322-1437
Online ISSN: 2345-2773
Publication License: 91/13216
Editorial Office Address: No.5, Saeedi Alley, Kalej Intersection., Enghelab Ave., Tehran, Iran,
P.O.Box: 13145-799 Tel: (+9821) 88930150 Fax: (+9821) 88930157
E-mail: info@jist.ir , infojist@gmail.com
URL: www.jist.ir

Indexed by:

- SCOPUS	www. Scopus.com
- Index Copernicus International	www.indexcopernicus.com
- Islamic World Science Citation Center (ISC)	www.isc.gov.ir
- Directory of open Access Journals	www.Doaj.org
- Scientific Information Database (SID)	www.sid.ir
- Regional Information Center for Science and Technology (RICeST)	www.ricest.ac.ir
- Iranian Magazines Databases	www.magiran.com

Publisher:
Iranian Academic Center for Education, Culture and Research **(ACECR)**

This Journal is published under scientific support of
Advanced Information Systems (AIS) Research Group and
Digital & Signal Processing Research Group, ICTRC

Acknowledgement

JIST Editorial-Board would like to gratefully appreciate the following distinguished referees for spending their valuable time and expertise in reviewing the manuscripts and their constructive suggestions, which had a great impact on the enhancement of this issue of the JIST Journal.

(A-Z)

- Alizadeh Noughabi, Havva, Islamic Azad University of Gonabad, Iran
- Ahmadizad, Arman, University of Kurdistan, Kurdistan, Iran
- Alavi, Seyed Enayatallah, Shahid Chamran University, Ahvaz, Iran
- Azizi, Sadoon, University of Kurdistan, Iran
- Asgari Tabatabaee, Mohammad Javad, University Of Torbat Heydarieh, Razavi Khorasan, Iran
- Agahi, Hamed, Islamic Azad University of Shiraz, Iran
- Babaei, Shahram, Islamic Azad University of Tabriz, Iran
- Darmani, Yousef, K. N. Toosi University of Technology, Tehran, Iran
- Ebadati, Omid Mahdi, Kharazmi University,Tehran, Iran
- Eshaghi, Mohadeseh, Islamic Azad University of Tehran, Iran
- Dehkharghani, Rahim, University of Bonab, East Azerbaijan, Iran
- Fadaeieslam, Mohammad Javad, Semnan University, Iran
- Farbeh, Hamed, Amirkabir University,Tehran, Iran
- Fakhar, Fatemeh, Payame Noor University, Ahvaz, Iran
- Farsijani, Hassan, Shahid Beheshti University, Tehran, Iran
- Farsi, Hassan, University of Birjand, South Khorasan, Iran
- Ghasemzadeh, Mohammad, Yazd University, Yazd, Iran
- Ghiamy, Mohammad Islamic Azad University of Ardabil, Iran
- Jamshidnejad, Amir, Academy of Gondishapur, Ahvaz, Iran
- Jazayeriy, Hamid, Babol Noshivani University of Technology, Mazandaran, Iran
- Kashef, Seyed Sadra, Urmia University, West Azerbaijan, Iran
- Kasaei, Shohreh, Sharif University, Tehran, Iran
- Khanmirza, Hamed, K. N. Toosi University of Technology, Tehran, Iran
- Mirzaei, Abbas, Islamic Azad University, Ardabil, Iran
- Mohammadzadeh, Sajad, University of Birjand, South Khorasan, Iran
- Momtazi, Saeed, Amir kabir University, Tehran, Iran
- Moradi, Gholamreza, Amirkabir University,Tehran, Iran
- Rasi, Habib, University of Technology, Shiraz, Iran
- Rezaei, Abdolhosein, Semnan University, Iran
- Soleimanian Gharehchopogh, Farhad, Islamic Azad University Urmia, Iran
- Shirvani Moghaddam, Shahriar , Shahid Rajaee Teacher Training University, Tehran, Iran
- Shamsi, Mahboubeh, University of Qom, Iran
- Sheykhpoor, Razieh, Ardakan University, Yazd, Iran
- Tanhaei, Mohammad, Ilam University, Ilam , Iran
- Yadollahzadeh tabari, meisam, Babol Islamic Azad University, Iran
- Yousefi, Saleh, Urmia University, West Azerbaijan, Iran

Table of Contents

A way to improve Adaptive Maintenance in Enterprise Architecture

Fereidoon Shams Aliee
Faculty of Computer Science and Engineering, shahid beheshti University,Iran
f_shams@sbu.ac.ir
safura oviesi*
Faculty of Computer Science and Engineering, shahid beheshti University,Iran
safura1371@gmail.com

Received: 24/Nov/2019 Revised: 16/Dec/2019 Accepted: 08/Feb/2020

Abstract
Lately, one of the biggest challenges in enterprise is the strategic alignment of information technology with business. Enterprises use various methods to achieve this strategic alignment. Enterprise architecture is an effective approach that allows optimal management of Enterprise's information technology and strategic alignment of IT functionalities and business requirements. These days, considering dynamic environments, there is the possibility of any kind of change in enterprise's conditions, especially after implementation of Enterprise architecture. This change of conditions demonstrates itself in a variety of forms. In these cases, decision making should be done as an appropriate response to these changes which should be predictable. Otherwise it would lead to lack of proper response and consequently, readjustment of Enterprise architecture document, which imposes many human and financial costs on the enterprise. In addition, the enterprise's progress toward competitive advantages will be stopped. The appropriate response and adaptability to these changes is a concept represented under the title of adaptive maintenance of Enterprise architecture. In the following article with the help of a software maintenance method and examining the possibility of extending into the Enterprise architecture maintenance, through a case study on Power Distribution Company of Golestan province, it is shown how the improvement of adaptive maintenance of Enterprise architecture can be assisted and the possibility of implementation of various types of changes in business and information technology without disturbing the Enterprise is provided.
Keywords: Enterprise architecture; strategic alignment; Enterprise architecture maintenance; Adaptability

1- Introduction

Enterprise architecture has three main phases as follows: [10]
Strategic planning of information technology
Planning of Enterprise architectural
Implementation of Enterprise architecture
Given that implementation of Enterprise architecture is not the last phase of Enterprise architecture, then, after implementation of Enterprise architecture, how can we respond to different types of changes and updates existing in Enterprise architecture, such that the Enterprise will always be up to date? that Answering this question and similar questions requires familiarization with a concept called adaptive maintenance which should be implemented in the Enterprise in order for it to have the capability of responding to different types of existing changes. However, the main problem is how this concept can be represented in Enterprise architecture. One of the best strategies to represent adaptive maintenance in Enterprise architecture is using the help of this concept in similar areas.[16] Given the age and diversity of software maintenance methods, this area can be selected as the closest area to Enterprise architecture maintenance. The difference between software architecture and Enterprise architecture is only in the scale and life span of their life cycle.[4] We are looking for a method that, while covering the concept of adaptability in the Enterprise, has the ability to be generalized to Enterprise architecture maintenance. Therefore, in the first step, the problem is whether software maintenance methods can be generalized to Enterprise architecture maintenance? If yes, how much is it? And, in the case of generalizability, how these methods can be used in order to satisfy maintenance capability in Enterprise architecture in the post-implementation phase?
One of the research gaps in this study and similar researches is the lack of attention to the impact of changes from higher layers on lower layers and only emphasize the impact of changes from lower layers on upper layers. In this paper, the main aim is providing a solution for the Adaptive maintenance of enterprise architecture. So far, studies have done this, but none of these studies have examined the adaptability of

enterprise architecture and the level of satisfaction with any changes and the impact of various changes in the enterprise And how adaptability and success rate or failure of the project by applying changes has not been addressed in a small way that we Let's solve all these challenges by generalizing the maintenance methods in software architecture.

In this study, we looking for answers to this question that How to generalize a software architecture maintenance method to organizational architecture so that we respond to different types of changes and updates existing in Enterprise architecture, such that the Enterprise will always be up to date? first, the studies conducted in the field of software architecture evaluation will be reviewed to select the appropriate method for generalization to enterprise architecture Finally, the proposed method will be proposed to solve the challenges in previous studies and then a case study will be examined in this paper to evaluate the proposed method The research data are examined by a tool called jucemnav in terms of the In terms of quantitative criteria of time and speed and qualitative criteria of accuracy, time, Error control, cognitive complexity with the help of the proposed method of this research and without the help of the proposed. Also, finally to review the results of this research and suggestions We will do future research in this area..

2- Review of literature and Research Method

2-1- Concept of Adaptive Maintenance in Enterprise Architecture

In Enterprise architecture, there is no resource that studies classification of different types of maintenance in detail. Most of the studies in the area of maintenance have looked at this feature generally, and refrained from addressing details and types of maintenance. For this reason, in Enterprise architecture, no special definition for adaptive maintenance was found. Therefore, considering the similarities between software architecture and Enterprise architecture which were mentioned before, redefinition of types of software maintenance in the area of Enterprise architecture will be as follows: With regard to the above, Enterprise architecture maintenance can be classified into three categories: corrective, adaptive, and perfective, which we will be redefined below in the area of Enterprise architecture: Corrective maintenance involves an operation that eliminates defective components of Enterprise architecture (replacing new business information systems instead of old systems).Adaptive maintenance operation involves the operations required to adapt the changes in each of different views of Enterprise architecture (the operation required to adapt to the increase of information systems in Enterprise).Perfective maintenance involves changes that are organized by user requests and lead to perfection of part or parts of the Enterprise (increasing and restoration of infrastructure layer of the Enterprise) **.**[2]

What is clear from the above definitions is the importance of adaptive maintenance, as in all three types of maintenance, directly and indirectly, the issue of adaptive maintenance has been addressed. In corrective maintenance also after elimination of defective components of Enterprise architecture, in case that there will be no appropriate adaptability to other components in order to find the proper alternative, then the Enterprise's architecture will face problems. In perfective maintenance also after the request by the user, in case that this request causes disorder in other processes or related components, this request will not be done and it requires adaptability of each request with other processes and components of Enterprise architecture, which itself, implies adaptive maintenance. For this reason, addressing adaptive maintenance is the basis for addressing other maintenance types that will be addressed in future researches. Therefore, in this study, focusing on adaptive maintenance, the basis will be provided for implementation of other maintenances in the Enterprise. [8]

The above definitions were redefinitions of types of software maintenance in the area of Enterprise architecture; but in order to make the concept of adaptive maintenance clearer, the definition of this type of maintenance according to software adaptive maintenance in the general scenario can be used which is as described below:[14]

Stimulation: adding, eliminating and changing implementation priority of each component existing in each view of Enterprise Architecture .[12]

- Stimulator: end-user, developer of Enterprise architecture, architect, CIO
- Environment: architecture maintenance and support phase
- View: business view, data architecture view, application architecture view, technology architecture view
- Product: Each of the different views of Enterprise architecture
- Possible responses: identification of any part affected by the changes; appropriate adaptability of the sectors engaged in each view of Enterprise architecture with the occurred change; level of satisfaction of Enterprise's ultimate goal from the extent of changes;

identification of the amount of effort needed to make the changes and the adaptability appropriate to them.
- Response criterion: importance, participation, and satisfactoriness of each view

2-2- Works Related to Software Maintenance

Among the software analysis methods, scenario-based methods are methods that through statement of functional states of system and examination of reactions specified in the architecture, try to analyze appropriateness of the architecture and the related qualitative features, including software maintenance .[15] There are several methods for scenario-based evaluation, each of which refers to different qualitative features. Table 1shows these methods and their applications better.

The three ALMA1, SAAM and ATAM2 methods are methods that address modifiability feature that, according to the proposed qualitative models below, is considered a criterion of maintenance capability; but among these methods, the ALPSM3 method, focusing on the qualitative feature of maintenance capability, is most consistent with the purposes of this research. So, through getting more familiar with this method, we try to generalize this method in Given Enterprise architecture. that these methods are specific to evaluation of software architecture, therefore, the techniques used in these methods are mostly unique to the software systems layer, and there is need for more studies beyond the scope of this study in order to determine whether they can be used in the Enterprise too.

2-2-1- ALPSM Method

ALPSM, due to its successful results, estimation of the effort required for repair and maintenance in the system in accordance with the built architecture, and estimation of the system's compatibility with scenarios based on prediction of maintenance capability, is considered a separate evaluation method and has several stages.This method takes the quadruple information including 1) features of requirements, 2) architectural design, 3) software engineers' expertise, and 4) historical data, as the inputs and provides a prediction of the average effort to maintain and repair the system. Through this method the scenarios for identification of maintenance requirements are aggregated and it helps architectural analysis for prediction of maintenance [2].This method is consisted of a series of change scenarios which are in fact a set of probable changes that are highly likely to occur and cause change in hardware or software. This method includes six essential steps as below:

1. Identification of the examined branch for Maintenance
2. Selection of scenarios
3. Evaluation of the weight of each scenario
4. Estimation of the size of the elements
5. Scripting scenarios
6. Calculation of the predicted effort for repairing and maintenance.

What was discussed above is in fact a comprehensive method for expressing software architecture maintenance. Although this method has merely been implemented for software architecture, however, given the proximity of its concepts to Enterprise architecture, it can be generalized to that too. In the proposed solution, ALPSM method will only be used for introduction and advancement of the research purposes so that in this way, while calculating the effort required for maintenance in Enterprise architecture, our main objective which is to improve adaptive maintenance in Enterprise architecture will also be obtained.

Table 1: Software scenario-based evaluation methods

Method	***SAAM [5]***	***ATAM [6]***	***SAAMER [11]***	***ALMA [8]***	***ALPSM***
Maturity stage	Refinement/ Dormant	Refinement	Dormant	Refinement/ Development	-
SA definition	In charge of the user	In charge of the user	In charge of the user	Not provided	-
Process support	Not clearly specified	Fully covered	Having a framework for process support	Incrementally implementing this capability	fully
Method purpose	Risk analysis	Analysis of sensitive and equilibrium points	Evaluation of soft architecture for evolution and capability of being reused	modifiability	Analysis of maintenance capability

[1] Architecture-level analysis of modifiability
[2] Architecture tradeoff analysis method
[3] Architecture Level Prediction of Software Maintenance

2-3- Previous research

2.3.1 Analysis of Variability of Enterprise Architecture:

Since variability is considered an obvious feature of adaptive maintenance, its analysis can play a very important role in guiding the method toward adaptive maintenance. In this study, software architecture evaluation methods have been used and it shows how these methods can help to analyze variability feature of Enterprise architecture. This method is a generalization of software architecture analysis method (SAAM). In this method [3], according to SAAM, firstly, a set of probable scenarios that the Enterprise may face in future are shared and then these scenarios are analyzed and prioritized. At this point, it uses BPMN in order to express and display business scenarios [9]. Then, in order to analyze the data, through ArchiMate language and other languages such as UML, it models other components such as data layers, information systems, and infrastructure. The purpose of this phase is to design the architecture such that it will be understandable for all stakeholders. Then it prioritizes direct and indirect scenarios and ultimately, all types of existing changes in these scenarios and the involved interactions are identified.

Given the involved components, the amount of required cost for variability per person per day is calculated based on the opinions of Enterprise experts.In this method; different modeling languages have been used to model the components of different layers. In addition, SAAM method used to generalize this method has not been used only for variability, but also has addressed other features too. This method shows that software architecture evaluation methods can be generalized to Enterprise architecture and therefore, it has an important role in proving the proposed method. An example of this modeling has been shown in Figure 1. This modeling does not specify the extent of the influence of change on other components of other layers. Moreover, adaptability method has not been shown in this method and only the type of scenario and the affected components have been specified.

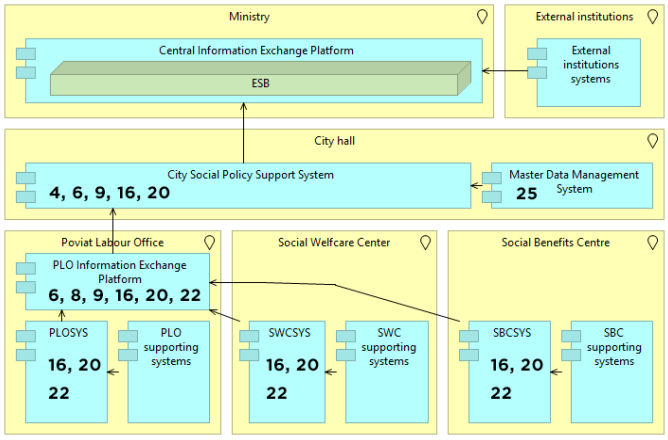

Fig..1 Modeling variability using ArchiMate language and specifying the components involved in each change scenario. [3]

2.3.2 A New Method Based on Genetic Algorithm and ATAM to Prioritize Enterprise Architecture Scenarios (Case Study: Enterprise Architecture Maintenance)

This method is one of the few methods that have used software evaluation methods in Enterprise architecture. In this study, ATAM has been used in order to analyze Enterprise architecture. This method uses different qualitative scenarios so that in this way, while identifying different decisions and compromising between them, will determine whether or not these decisions, given the architectural structure, are consistent with other qualitative features? [7] After prioritizing scenarios by ATAM, an appropriate scenario is found among the prioritized scenarios through genetic algorithm. This method examines maintenance and interactivity with the help of ATAM method and genetic algorithm. In this method, firstly, the indicators of maintenance are prioritized by the stakeholders; then the change scenarios for maintenance and interactivity are specified and prioritized by the stakeholders. After this stage, the degree of influence of each scenario on the qualitative feature is determined through quantitative values,[13] and finally, these prioritized values are injected as inputs into genetic algorithm, and after implementation of genetic algorithm in MATLAB software, the scenarios with the highest priority and the most appropriate scenario are selected as the desired change scenario, so that in this way, the cost and time of Enterprise maintenance will be reduced.

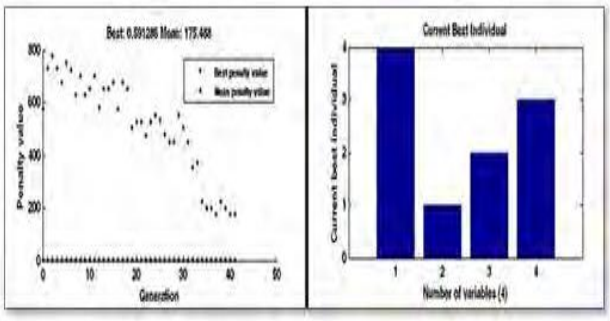

Fig 2 Display and selection of the scenario appropriate for change in MATLAB tool with the help of genetic algorithm.[7]

This method, despite being a very new idea in Enterprise architecture and reducing complexity of implementation time, has not proposed anything about the affected components and how to deal with these change scenarios.

2-4- Challenges

All of the methods mentioned in the previous section examine Enterprise architecture maintenance. In most of these methods, change scenarios have been considered as the input of the proposed method, which is itself the main incentive of the present research in order to use scenario-based methods. But the problem which can be seen in most of these methods, despite very high importance of business layer, is very little attention to this layer, since these researches have mostly focused on information systems layer rather than business layer. In addition, in these methods special attention has been made to selecting appropriate scenario for Enterprise architecture maintenance, but little attention has been paid to how to apply this scenario, identification of involved components, the amount of influence on other components, and adaptability method. In this study we are seeking a strategy through which, while selecting the appropriate scenario for maintenance, the other existing challenges can also be removed.

3- The Proposed Method

The proposed method, in addition to providing a solution to improve adaptive maintenance, calculates the effort required for adaptive maintenance in Enterprise architecture. It should be noted that given the wide range of Enterprise architecture and impossibility of examining all its layers in this study, only two important and challenging layers of business and information systems are examined. Also, the data layer which is located between these two important layers will be examined as Enterprise decisions, and examination of other layers will be part of the proposed works. This method has been expressed by generalization of software maintenance and ALPSM method. The proposed method has largely been generalized from ALPSM method and is consisted of four stages that the first three ones address improvement of adaptive maintenance of Enterprise architecture and the last stage estimates the effort required to maintain Enterprise architecture. Figure 3 shows the stages of the proposed method, along with its inputs and outputs.

Inputs:

- Developed Enterprise architecture
- Change scenarios

Method:

1. Description and modeling of the desired architecture
2. Prioritization of change scenarios
3. Examination and adaptability of change scenarios
4. Estimation of the effort needed for adaptive maintenance

Outputs:

- Modeling of Enterprise architecture
- Identification of changes in Enterprise architecture layers
- Responsiveness and adaptability
- Calculation of the effort needed for maintenance

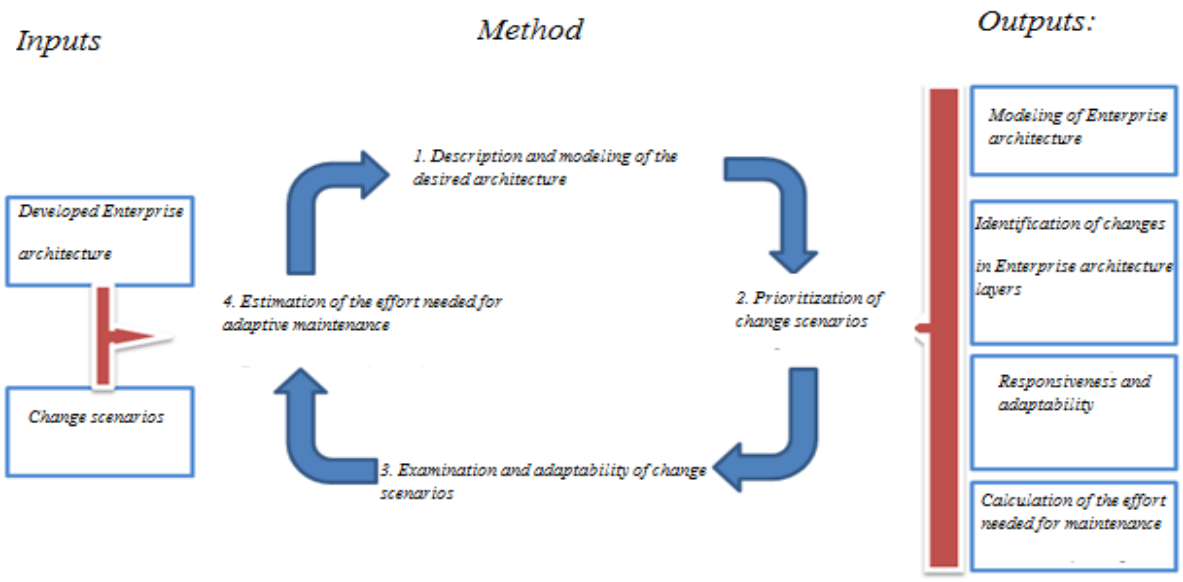

Fig. 3: Proposed method for improving adaptive maintenance in Enterprise architecture

Since the methods proposed in the area of Enterprise architecture maintenance in fact predict this qualitative feature, according to previous researches, four criteria of accuracy, cost of using the model, cognitive complexity and error control can be considered for evaluation of prediction models.

The models built for other qualitative features have been examined in terms of these four criteria. In this source, the following seven characteristics have been described in order to meet these four criteria:

1. Formal prediction theory
2. Automatic prediction
3. The rules set for the considered qualitative feature
4. Clarity and fluency of the formal method used
5. The reusable prediction theory
6. Prediction based on probabilities
7. Model abstraction

3-1- Description and Modeling of the Desired Enterprise Architecture

3.1.1 Description of the Desired Enterprise Architecture

At this stage, we will select and determine the layers of Enterprise architecture in which improvement of adaptive maintenance is going to be done.

3.1.2 Modeling of the Described Enterprise Architecture

In the second part of the first stage, the Enterprise architecture described in the previous part is modeled. The modeling language of URN and JUCMNAV tool are used for modeling [1].This modeling has been implemented down to top. The changes occur in lower layers, and then the impact of changes is transferred to higher layers of Enterprise architecture.

The advantage of using this modeling is using three very important parameters of contribution level, importance level, and satisfaction level of the elements used in the model, which is considered the strength of the proposed method. According to Enterprise experts, these three parameters are in fact considered three key parameters for

adaptability; and after any change and the adaptability appropriate to that, these three are among the most used parameters that must be examined. The contribution level parameter is based on collaborative links between the internal modeled elements and shows the amount of contribution of the elements in the lower layer to the elements in the higher layer in order to achieve main objective of the Enterprise. The importance level parameter is located inside each element and expresses the importance of each element among other elements involved in the same layer. The satisfaction level parameter shows level of satisfaction of the high level layer element in the face of changes applied to low level layer elements. In order to calculate the first two

parameters, namely, the contribution level and the importance level of each element, AHP method is used, whose inputs (prioritization of elements of each layer) are obtained through Enterprise experts. In order to calculate the third parameter, namely satisfaction level, Bottom up propagation algorithm is used which has been written by Amyot et al. If the element X is located in the layer higher than A, B, C elements with the contribution level of a, b, c and satisfaction level of d, e, f, then, satisfaction level of the element X(sle(X)) is obtained using Formula 1:

$$sle(X)=(a*d)+(b*e)+(c*f) \quad (1)$$

In general, in this section, firstly each layer of Enterprise architecture described in the first section is modeled separately according to the above points. Then, the layers are linked to each other through cooperative links, and finally, correctness of this relationship is examined with the help of adaptation rules. After automatically examining the rules, the desired algorithm is executed so that satisfaction level of the elements in the higher layers, and ultimately, the Enterprise's goal, in the conducted modeling, will be determined.

3.1.3 Prioritization of Change Scenarios

At this stage, various types of changes that may occur in the Enterprise architecture layers are examined in the form of change scenarios. These changes are categorized into three categories of simple, incremental and decreasing changes:

- Simple changes are the changes that affect implementation priority of Enterprise architecture elements, goals, functions, and information systems.
- Incremental changes are changes that lead to addition of an Enterprise element, goals, functions, and information systems in the desired layers.
- The decreasing changes are changes that are done as the result of elimination and reduction of Enterprise architecture elements, goals, functions, and information systems in the desired layers

At this stage, different change scenarios in the desired layers are examined and, similar to ALPSM method, they are prioritized based on the opinions of Enterprise experts. Determining these scenarios is the responsibility of the Enterprise's architect.

3.1.4 Examination and Adaptability of Change Scenarios

At this stage, which is considered the most basic stage in this method, we will examine how the model reacts and adapts to the applied changes? As previously stated, this adaptability will be done in the face of simple, incremental, and decreasing changes in the desired layers which are business layer and information systems in this research.

3.1.5 Estimation of the Effort Required for Adaptive Maintenance

At this stage, with the help of the models built before and after the changes, the amount of effort required for adaptive maintenance can be obtained. In order to achieve this very important formula, the total number of people involved in the Enterprise must be determined. In the model implemented in this research in the case study of section 5, individuals or actors are in fact parts of the three layers of information systems, business, and decision-making that contribute to achieving Enterprise goal. In this research, the parameters determined for each actor (part) and indicated in the desired modeling include the level of satisfaction from each actor based on satisfaction from internal elements of each actor; however, in order to obtain the effort required for adaptive maintenance based on person/day, the very important parameter based on which the individuals' allocation is made is importance level parameter. According to a survey conducted on the Enterprise experts, the importance level parameter with 49% was selected as the highest factor in allocation of forces to each sector, and the parameters of contribution level and satisfaction level with 30% and 21% were placed in the second and third ranks, respectively.

According to the opinion of Enterprise experts and the Enterprise's architect, in order to allocate people to each unit, the importance level of each unit plays a very significant role in allocation of individuals, because the more important each unit is, the more employees it requires; while, according to its importance, it has more contribution, directly or indirectly, in achieving Enterprise goal, while the level of satisfaction from the unit has a lower relation to the employees allocated to that unit. For this reason, in this section we need to know the importance level of each actor (part), which this is also obtained through binary AHP comparison.

Firstly, the number of people required for each actor (part) is determined through the following formula:

$$N(\text{actor}) = p(\text{actor}) * N\text{total} \quad (2)$$

$$P(\text{actor}): \text{Importance level of each actor} \quad (3)$$

Each actor has a number of internal elements that are shown in the form of goal, information system, or decision (tasks). In order to obtain the individuals involved in each internal element, the following formula is used:

$$N(\text{initial element}) = N(\text{actor}) * p(\text{initial element}) \quad (4)$$

These formulas help us obtain the number of people involved in each unit and each goal or information

system or each decision making. After the changes are made, these formulas are re-calculated and the number of people based on each change is obtained; Finally, the following formula is used in order to obtain the effort required for adaptive maintenance:

$$E(EAAM)=N(ChangeElement) + \sum_{I=1}^{N\ (ChangeElement)} N\ (The\ effectedElements) \quad (5)$$

- E (EAAM): The effort required for adaptive maintenance of Enterprise architecture per person per day
- N (ChangeElement): The number of people involved in changing the desired element
- N (The effectedElements): The number of people involved in the elements affected by the created change

4- Evaluation of the Proposed Method

The proposed method was presented with the subject of providing a method for improving adaptive maintenance of Enterprise architecture. There are several methods for evaluation, and each of these methods is used for a specific application. Among these methods, the most appropriate evaluation solution for the proposed method is using of a real sample. One of the most effective ways in order to indicate improvement in adaptive maintenance of Enterprise architecture is to compare the real sample with the sample implemented with the help of the proposed method; and case study allows for access to this method. This case study, as Enterprise architecture has been fully developed and implemented in that, is considered a very good case for this research. This company due to having a dynamic environment is very suitable for examination of adaptive maintenance. Enterprise architecture documentations for Power Distribution Company of Golestan province have been provided by Golestan Software Management Group and it includes the architecture of the existing situation and the architecture of the favorable situation. In the following, the steps of the proposed method will be evaluated:

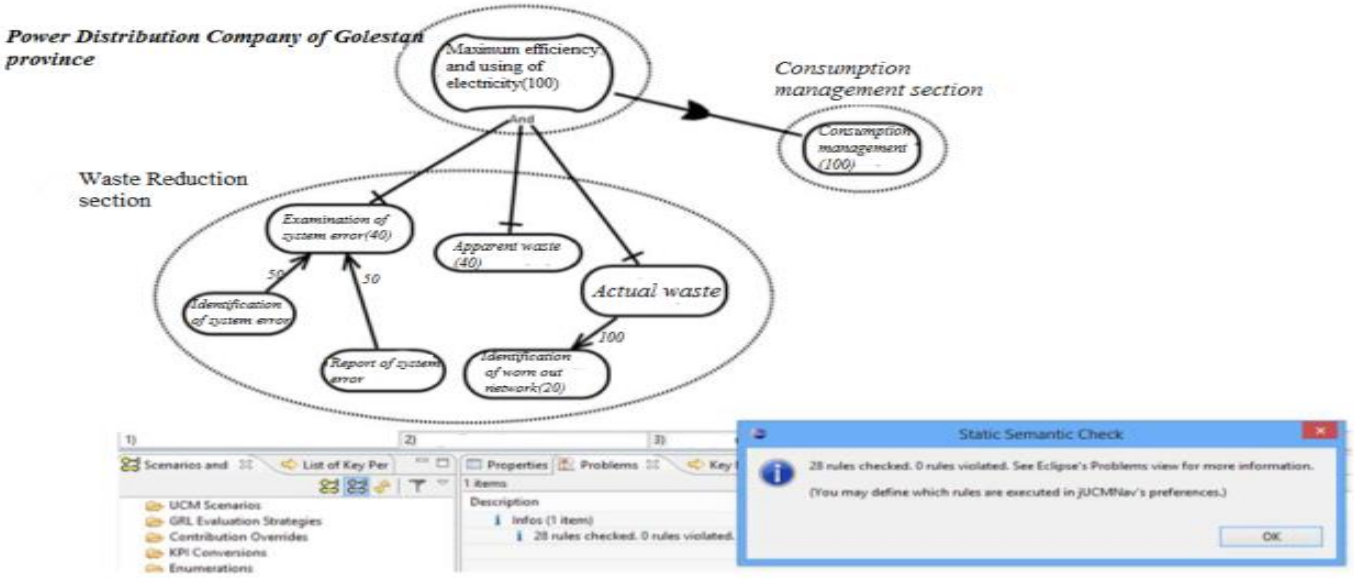

Fig. 4 Examination of correctness of the levels implemented by Power Distribution Company of Golestan province

4-1- Description and Modeling of the Desired Architecture

Power Distribution Company of Golestan province: Maximum efficiency and using of electricity
Consumption management section: Consumption management
Waste Reduction section:
Examination of system error/ Apparent waste/ Actual waste
Identification of system error/ Report of system error/ Identification of worn out network.
The implemented model is a bottom-up model and satisfaction levels of the modeled structure .have been developed with the help of bottom-up algorithm. The initial valuation of information system indicators is obtained by specifying the 4 values of goal, the mean, the worst, and the assessment mode that has been valued by the experts of Power Company. For example, indicators of PM system have been examined and evaluated as a sample (Table 2). Then, these indicators, based on the evaluated value, are implemented by the developed algorithm in order to obtain satisfaction levels of the desired indicator. These indicators, with the help of the formula proposed in the desired algorithm, transfer satisfaction levels to the level of information systems and then to higher levels (Figures 5).

4-2- Prioritization of Change Scenarios

Change scenarios include three types of deletions, additions, and simple changes to the implementation priorities of internal elements and links. In order to select scenarios, we need to implement scenarios that have actually been implemented in this company so that when calculating adaptive maintenance of Enterprise architecture of this company, we can make an acceptable comparison between the actual amount of maintenance and the maintenance value obtained by the proposed method. One of the scenarios that had occurred in this company and had failed due to lack of sufficient understanding of adaptability issue was removing GIS information system and integrating it with System 121. Despite the fact that this scenario could have had positive impacts on the Enterprise, it has failed due to lack of appropriate adaptability and lack of adequate understanding of the positive effects of this change on other levels.

4-3- Examination and Adaptability of Change Scenarios

- Removing GIS information system: GIS system is one of the most important systems used in Power Distribution Company. With a brief looking at GIS capabilities, it is seen that one of the mostly used GIS capabilities is identifying worn-out networks, fixing network defects, and easier decision making about network which plays a significant role in preventing power losses. After removing GIS information system, with the help of AHP paired comparison, the model will be adapted and the effect of this change is applied to the higher levels.

By removing this information system and integrating it with System 121, we will have the amount of satisfaction level increased on other levels which can be seen in the Figure.

Table 2: Valuation of PM information system indicators

	Goal	***Mean***	***The worst***	***Evaluation***
Relation with other units	90	70	50	72.5
Information quality	90	70	50	80
Service quality	90	70	50	90
Encapsulation level	90	70	50	65
Satisfaction of the user from the system	90	70	50	80
Cost	90	70	50	75

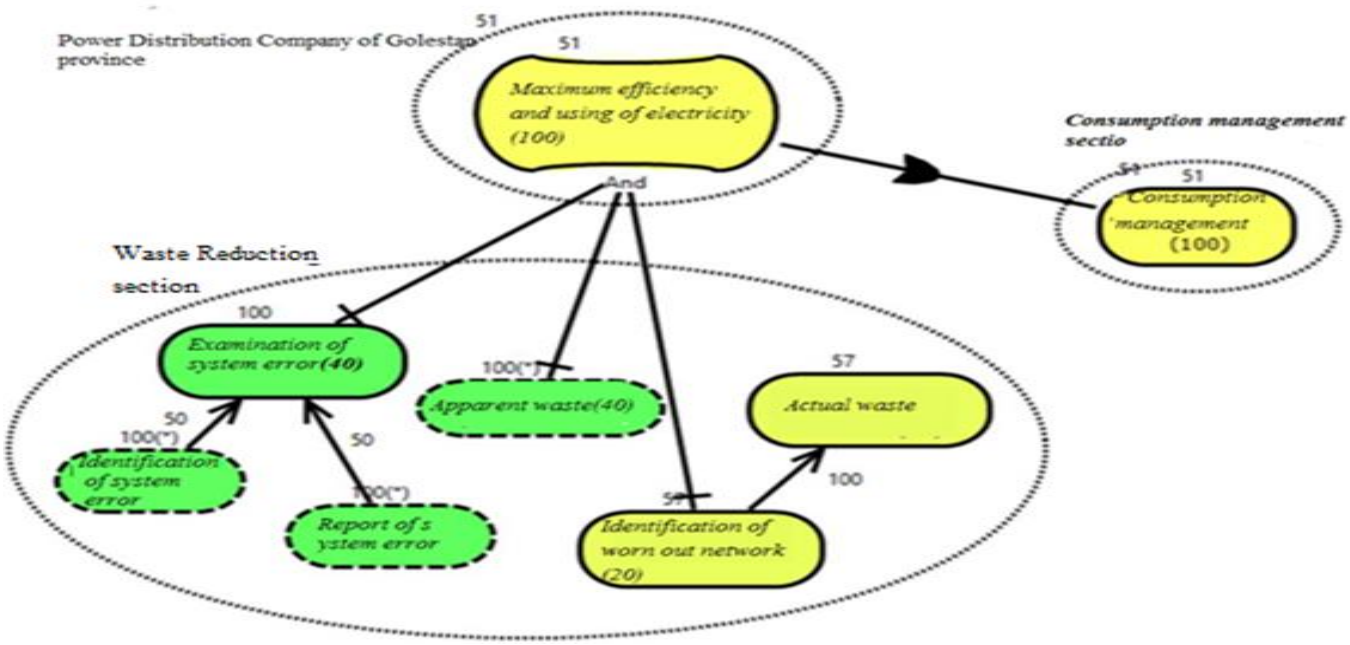

Fig.5 Implementation of the algorithm on business level in order to obtain satisfaction levels

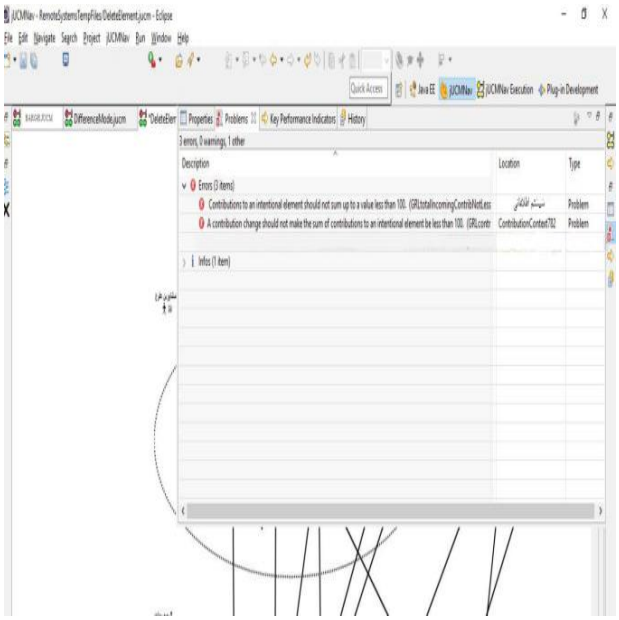

Fig.6 How to fix the errors occurred due to change

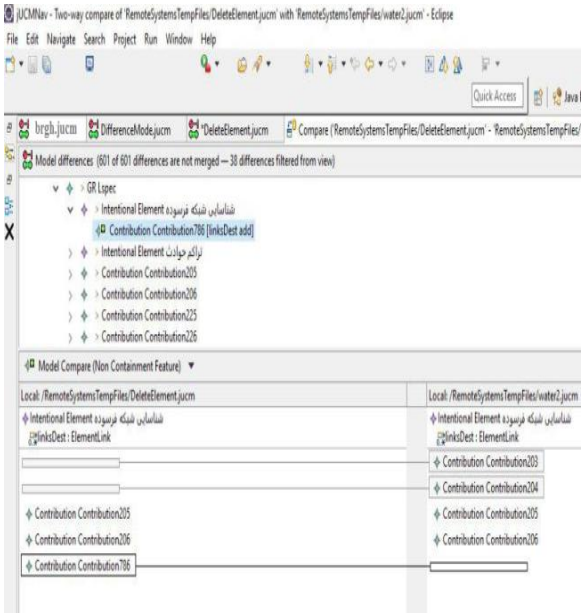

Fig. 7 Showing the elements of other levels before and after removal of the GIS information system

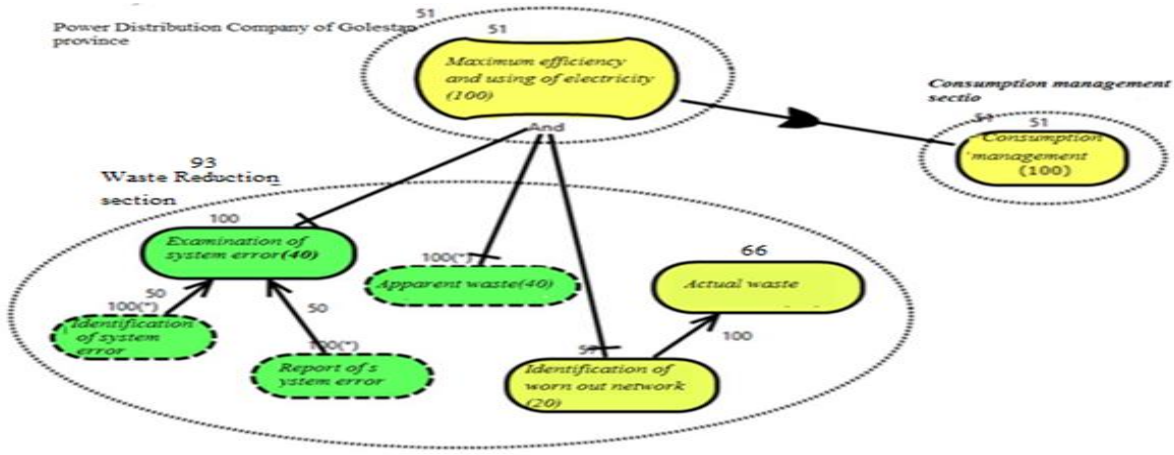

Fig. 8: The impact of removal of GIS information system on business level

- Calculation of adaptive maintenance of change scenarios: In order to calculate adaptive maintenance, firstly the number of people involved in each section must be identified. The total staff involved in this target are 145 people who must be distributed among various actors (units or sectors including goals, tasks, or information systems), and each actor, based on higher importance level, allocates more people to itself. Actors who are involved in this goal include project consultants, waste reduction section, subscribers and branch section, utilities section, consumption management section, information systems section and technical engineering section. Using AHP method, the importance level of each actor is determined, and then the number of people allocated to that is specified. With the help of the obtained figures, given that the total number of actors is 145 people, the number of individuals related to each actor is determined. Now, using the number of people obtained for each actor, the number of people involved in each target, sub-target, task, and information system are determined.

Now, having detailed information about the number of actors in each target, activity and task, the amount of adaptive maintenance is calculated per person per day.

$$E(EAAM)=N(ChangeElement) + \sum_{I=1}^{N\ (ChangeElement)} N\ (The\ effectedElements) \quad (6)$$

According to this formula, firstly the number of people involved in the considered change which is removal of GIS information system must be identified, that the number of these individuals is limited to 6 people. The number of people involved in sectors affected by this change is obtained. According to the mentioned interpretations and Formula One, adaptive maintenance of Power Distribution Company of Golestan province in case of removal of GIS information system per person per day is: 74 + 6 = 80 person per day .While according to the information obtained from Power Distribution Company, before implementation of the proposed method, due to this change in the company, the amount of adaptive maintenance of this company had reached 113 person per day, that due to lack of sufficient knowledge about the affected elements and lack of appropriate adaptability, about 33 people had been used inappropriately and extremely compared to actual maintenance of the Enterprise; that this is a significant amount.

4-4- Qualitative Evaluation of the Proposed Method

Since the methods proposed in the area of Enterprise architecture maintenance in fact predict this qualitative feature, according to previous researches, four criteria of accuracy, cost of using the model, cognitive complexity and error control can be considered for evaluation of prediction models. The models built for other qualitative features have been examined in terms of these four criteria. In this source, the following seven characteristics have been described in order to meet these four criteria:

1. Formal prediction theory
2. Automatic prediction
3. The rules set for the considered qualitative feature
4. Clarity and fluency of the formal method used
5. The reusable prediction theory
6. Prediction based on probabilities
7. Model abstraction

The first feature results in higher accuracy of the model. Since our model rules are normalized as OCL, there is no possibility of misinterpretation, and this helps higher accuracy of the model. On the other hand, these formal rules can be used for implementing the method through tools, that this reduces the cost of using the method and human error and results in higher accuracy in prediction. In addition, the model has been implemented completely separate and leveled. This leads to easy use of the model and reduction of its learning curve, which will ultimately lead to positive impact on cognitive complexity of the model. In order to increase accuracy, the formal rules used in this model are needed to be sufficiently clear and fluent and can describe the problem space clearly. For example, in the area of maintenance, there are many concerns about examination of adaptability. For this purpose, the used formal method must allow description of adaptive architectural structure.

Using of OCL language, by expressing the rules, allows for query about the model and use of logical calculations and operators which provide the analyst with sufficient power. It also reduces the cost of using the model due to not needing a specialist in the field and reduced its cognitive complexity, because the analyst only needs to focus on modeling and does not need to define relationships in each scenario, and the formal rules have this responsibility. Generally, the following four criteria are shown in this method as follows:

- Accuracy: Accuracy of the proposed method depends on data collection method. If the analyst determines the relationships between the levels of Enterprise architecture and model elements based on his own judgment, this accuracy will be low, and if data collection methods based on literature review, data mining, or expert surveys are used, more accurate results can be expected. So, in this method, due to the use of research literature, views of Enterprise experts and the use of OCL adaptable rules, we have a higher accuracy.
- Cost: The use of the proposed method, due to its lack of complexity, does not require a very high cost; in addition, this method, after one time of implementation, provides the analysts with the reusable results.
- Cognitive complexity: The use of the proposed method, due to determining clear and specified steps, is estimated to be relatively simple. Since this method has been generally presented, it can also be used for adaptability of other qualitative features and the learning chart can be expected to be short.
- Error control: In the compromise between accuracy and cost, in the information collection phase the researcher can use available resources, such as personal opinion, subject literature, data mining, and expert survey methods. It is clear that accuracy of the model will be higher based on the amount of effort spent on collection of basic information (cost).

5- Discussion and Conclusion

In this research, the main question of the research is How to generalize a software architecture maintenance method to Enterprise architecture so that we respond to different types of changes and updates existing in Enterprise architecture, such that the Enterprise will always be up to date? answered as follows, and finally, the identification of different types of changes in the enterprise. The effect of lower layer changes on top of how to deal with these changes and adaptability to each change was successfully done by examining a sample. The actual reduction in work time and cost was significantly reduced with the proposed method.

The first step in ALPSM method selects the maintenance branch, that in Enterprise architecture maintenance method, adaptive maintenance branch was selected. In the second and third stages, the scenarios are selected and weighed in order for prioritization, that in the proposed method, through survey of the Enterprise experts, in the second stage, the scenarios are prioritized and scenarios with the highest priority are selected. In the fourth stage, the amount of the component which is in fact the amount of effort for changing the desired component is estimated, that in the proposed method, using importance level parameter and determining the people specific to each Enterprise component at the scenario examination and adaptability stage, this amount will be obtained. Also, in the fifth stage, ALPSM scripts the proposed scenarios so that in this way the involved components and the impact of the change on other components can be determined, that in the proposed method, using the parameters of contribution level, importance level, satisfaction level, and developed algorithm and adaptation rules, the components affected in the related layers and the amount this effect are easily determined, and finally, in ALPSM method, the effort required for maintenance is estimated, that given the size of Enterprise architecture compared to software architecture and differences in some calculations, the effort required for Enterprise architecture maintenance is obtained given the specified parameters.

Table3 :Comparison of the proposed method with previous studies

Calculation of Maintananc	***Display the level of satisfaction with adaptability***	***Adaptability***	***Display the affected components***	***Select the appropriate change scenario***	***Studies***
×	×	×	✓	✓	[3]
×	×	×	×	✓	[7]
×	×	×	✓	✓	[14]
✓	✓	✓	✓	✓	Suggested method

6- Recommendations for Future Research

As was noted in this study, this research despite the possibility of covering all layers only examines the two layers of information systems and business systems that future studies can investigate other layers.

1. Adaptability rules in this study are specific to the two layers of business and information systems that one of the other things that can be done in future is to write general adaptable rules that can be applied to all Enterprise architecture layers.
2. This method only shows the impact of bottom-up changes. In future studies, using an algorithm that can show the impact of changes in both bottom-up and up-bottom sides can play a significant role in advancement and development of this research.

References

[1] D.Amyot,"Consistency Analysis for User Requirements Notation Models",Ninth International i* Workshop (iStar), At Beijing, China, Volume: CEUR-WS Vol-1674,2016.

[2] P.Bengtsson," Architecture Level Prediction of Software Maintenance", Department of Software Engineering aComputerScience University of Karlskrona/Ronneby S-372 25 Ronneby Sweden,1999.

[3] NR.Buschand and A.Zalewski," Enterprise Architecture Modifiability Analysis". In Towards a Synergistic Combination of Research and Practice in Software Engineering, Springer, Cham, pp. 119-134, 2018.

[4] Chief Information Officer Council Version 1.0,A Practical Guide to Federal Enterprise Architecture, February 2001, [Online]. Available: https://www.gao.gov/.

[5] R.KazmanBass and L.Abowd and M.Webb, "SAAM: A method for analyzing the properties of software architectures". In Proceedings of 16th International Conference on Software Engineering, IEEE, pp. 81-90,1994.

[6] R.Kazman and M. Klein, and P. Clements," ATAM: Method for architecture evaluation",Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst,2004.

[7] M.Karimi AND S.M.Sharafi and M.N .Dehkordi, "A new approach based on genetic algorithm for prioritizing quality scenarios in enterprise architecture evaluation. International Journal of Computer Science Engineering ", IJCSE, 3(01), 21-31, ,2014.

[8] D.Amyot,"Consistency Analysis for User Requirements Notation Models",Ninth International i* Workshop (iStar), At Beijing, China, Volume: CEUR-WS Vol-1674,2016.

[9] P.Bengtsson," Architecture Level Prediction of Software Maintenance", Department of Software Engineering aComputerScience University of Karlskrona/Ronneby S-372 25 Ronneby Sweden,1999.

[10] NR.Buschand and A.Zalewski," Enterprise Architecture Modifiability Analysis". In Towards a Synergistic Combination of Research and Practice in Software Engineering ,Springer, Cham, pp. 119-134,2018.

[11] Chief Information Officer Council Version 1.0,A Practical Guide to Federal Enterprise Architecture, February 2001, [Online]. Available: https://www.gao.gov/.

[12] R.KazmanBass and L.Abowd and M.Webb, "SAAM: A method for analyzing the properties of software architectures". In Proceedings of 16th International Conference on Software Engineering, IEEE, pp. 81-90,1994.

[13] R.Kazman and M. Klein, and P. Clements," ATAM: Method for architecture evaluation",Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst,2004.

[14] M.Karimi AND S.M.Sharafi and M.N .Dehkordi, "A new approach based on genetic algorithm for prioritizing quality scenarios in enterprise architecture evaluation. International Journal of Computer Science Engineering ", IJCSE, 3(01), 21-31, ,2014.

[15] D.Amyot,"Consistency Analysis for User Requirements Notation Models",Ninth International i* Workshop (iStar), At Beijing, China, Volume: CEUR-WS Vol-1674,2016.

[16] P.Bengtsson," Architecture Level Prediction of Software Maintenance", Department of Software Engineering aComputerScience University of Karlskrona/Ronneby S-372 25 Ronneby Sweden,1999.

[17] NR.Buschand and A.Zalewski," Enterprise Architecture Modifiability Analysis". In Towards a Synergistic Combination of Research and Practice in Software Engineering ,Springer, Cham, pp. 119-134,2018.

[18] Chief Information Officer Council Version 1.0,A Practical Guide to Federal Enterprise Architecture, February 2001, [Online]. Available: https://www.gao.gov/.

[19] R.KazmanBass and L.Abowd and M.Webb, "SAAM: A method for analyzing the properties of software architectures". In Proceedings of 16th International Conference on Software Engineering, IEEE, pp. 81-90,1994.

[20] R.Kazman and M. Klein, and P. Clements," ATAM: Method for architecture evaluation",Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst,2004.

[21] M.Karimi AND S.M.Sharafi and M.N .Dehkordi, "A new approach based on genetic algorithm for prioritizing quality scenarios in enterprise architecture evaluation. International Journal of Computer Science Engineering ", IJCSE, 3(01), 21-31, ,2014.

[22] D.Amyot,"Consistency Analysis for UserRequirements Notation Models",Ninth International i* Workshop (iStar), At Beijing, China, Volume: CEUR-WS Vol-1674,2016.

[23] P.Bengtsson," Architecture Level Prediction of Software Maintenance", Department of Software Engineering aComputerScience University of Karlskrona/Ronneby S-372 25 Ronneby Sweden,1999.
[24] NR.Buschand and A.Zalewski," Enterprise Architecture Modifiability Analysis". In Towards a Synergistic Combination of Research and Practice in Software Engineering ,Springer, Cham, pp. 119-134,2018.
[25] Chief Information Officer Council Version 1.0,A Practical Guide to Federal Enterprise Architecture, February 2001, [Online]. Available: https://www.gao.gov/.
[26] R.KazmanBass and L.Abowd and M.Webb, "SAAM: A method for analyzing the properties of software architectures". In Proceedings of 16th International Conference on Software Engineering, IEEE, pp. 81-90,1994.
[27] R.Kazman and M. Klein, and P. Clements," ATAM: Method for architecture evaluation",Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst,2004.
[28] M.Karimi AND S.M.Sharafi and M.N .Dehkordi, "A new approach based on genetic algorithm for prioritizing quality scenarios in enterprise architecture evaluation. International Journal of Computer Science Engineering ", IJCSE, 3(01), 21-31, ,2014.
[29] S.R.Khayyami,." Evaluation and analysis of Enterprisearchitecture",Ph.D.dissertation,University of Shiraz,iran, , 2009.
[30] R. Lagerström, "Analyzing system maintainability using enterprise architecture models," jEA 2007, pp. 31-40, 2007.
[31] M. Lankhorst," Enterprise Architecture at Work: Modeling, Communication and Analysis," 3th ed. Springer, 2013.
[32] M .M .Lehman," Implications of Evolution Metrics on Software Maintenance," Department of Computing Imperial College London SW7 2BZ,2015.
[33] L. Manzur and J. Mario Ulloa, M. Sanchez and J. Villalobos, "xArchiMate: Enterprise Architecture simulation, experimentation
[34] and analysis," Simulation-Transactions of the Society for Modeling and Simulation International, 2015 ,vol. 91, no. 3, pp. 276-301.
[35] P. Narman and P. Johnson,"Enterprise Architecture: A Framework Supporting System Quality Analysis," presented at the 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), Annapolis, MD, USA, 2007.
[36] M. Razavi Davoudi and K. Sheikhvand, "An Approach towards Enterprise Architecture Analysis,using AHP and Fuzzy AHP, "2012.
[37] M.Razavi ," Presentation of a new framework for analysis of qualitative features of Enterprise architecture," PhD.dissertation ,Tehran University of Science and Technology,iran, 2009.
[38] F. Shams and A. Mahjourian," Introduction to the principles and basics and architectural methods of service-oriented Enterprise," Shahid Beheshti University, Tehran,iran, 2015.

Fereidoon Shams Aliee is a full-time Assistant-Professor of Computer Engineering in the Faculty of Computer Science and Engineering at shahidbeheshti University, Iran. He received his Ph.D. degree in Doctor of Philosophy, from Department of Computer Science, Manchester University, UK,1996.. He is conducting research activities in the areas of Software Architecture, Enterprise Architecture, Service Driven Architecture, Agile Methodologies, Ultra-Large-Scale (ULS) Systems, Ontological Engineering.

safura oviesi received the B.S. degree in information tecnologhy Engineering from sistan and baluchestan university, Iran in 2014 and M.S. degree in information tecnologhy Engineerin-Enterprise Architecture from shahidbeheshti university,Tehran,Iran, in 2018. Her research interests include Information security, Enterprise Architectures,processes, mining and Data mining

Density Measure in Context Clustering for Distributional Semantics of Word Sense Induction

Masood Ghayoomi
Faculty of Linguistics, Institute for Humanities and Cultural Studies, Tehran, Iran
m.ghayoomi@ihcs.ac.ir

Received: 22/Nov/2019 Revised: 17/Dec/2019 Accepted: 8/Feb/2020

Abstract
Word Sense Induction (WSI) aims at inducing word senses from data without using a prior knowledge. Utilizing no labeled data motivated researchers to use clustering techniques for this task. There exist two types of clustering algorithm: parametric or non-parametric. Although non-parametric clustering algorithms are more suitable for inducing word senses, their shortcomings make them useless. Meanwhile, parametric clustering algorithms show competitive results, but they suffer from a major problem that is requiring to set a predefined fixed number of clusters in advance.
The main contribution of this paper is to show that utilizing the silhouette score normally used as an internal evaluation metric to measure the clusters' density in a parametric clustering algorithm, such as K-means, in the WSI task captures words' senses better than the state-of-the-art models. To this end, word embedding approach is utilized to represent words' contextual information as vectors. To capture the context in the vectors, we propose two modes of experiments: either using the whole sentence, or limited number of surrounding words in the local context of the target word to build the vectors. The experimental results based on V-measure evaluation metric show that the two modes of our proposed model beat the state-of-the-art models by 4.48% and 5.39% improvement. Moreover, the average number of clusters and the maximum number of clusters in the outputs of our proposed models are relatively equal to the gold data.

Keywords: Word Sense Induction; Word Embedding; Clustering; Silhouette Score; Unsupervised Machine Learning; Distributional Semantic; Density.

1- Introduction

Language is a means of communication to transfer a concept from a producer (speaker) to a recipient (listener). de Saussure [1] believes that language is composed of 'form' and 'meaning'. 'Form' which is tangible and recordable can be represented by a phonological or orthographic system; and 'meaning' which is abstract is very difficult to capture. The general learning process of humanbeings is that they use inductive methods to cluster the information in the brain by finding similarities and dissimilarities of instances. Through this clustering approach, concepts are discovered and words' meanings (senses) are found out. For instance, the inductive process of human's brain puts the word '*bank*' in two clusters due to having two different meanings, which are '*the financial place*' and '*near the river*'. Artificial intelligence and natural language processing aim at simulating this ability of human on a machine to learn a natural language. Word Sense Induction (WSI) is a task that causes a machine to induce word senses automatically from a raw data without using a prior knowledge or annotated data.

Providing knowledge sources of words' senses for machines, such as the WordNet [2] and OntoNotes [3], is normally done manually. There are drawbacks for this approach. Providing such data is expensive in terms of both time and cost and due to the language change over time, updating the lexical resource and revising it require additional cost and time. The number of senses of an existing word is not consistent; therefore, a new sense may be added to an existing word, or the senses of an existing word may become outdated. Moreover, senses may change with respect to the domain. To overcome the drawbacks, WSI can be a solution to provide helpful information for different tasks, such as machine translation and information retrieval.

Context clustering is one of the well-known successful approaches for WSI [4, 5, 6]. One challenge of this approach is choosing the appropriate clustering algorithm. The proposed models in the literature have benefited two important clustering approaches, namely parametric and non-parametric. Parametric clustering algorithms, such as partitioning K-means algorithm, require a predefined fixed number of clusters as an input parameter; while non-parametric clustering algorithms, such as Chinese Restaurant Process (CRP) [7] and Density-Based Spatial Clustering of Applications with

* Corresponding Author

Noise [8], make decision to define new clusters. Although non-parametric clustering algorithms are more suitable for inducing word senses, the reported results in the literature show the superiority of parametric clustering in the field. It has to bear in mind that parametric clustering suffers from the problem of requiring a predefined number of clusters. In this research, we aim at addressing this problem by capturing the optimum number of clusters using a density measure. To reach the goal, we use the parametric clustering approach for the WSI task and try to solve its major problem by utilizing the internal evaluation score defined for measuring the density of clusters.

Another challenge of the context clustering approach is the method to be used to capture the contextual information of words to be able to decide about their senses and to achieve accurate results. Distributional semantic representation of words, such as Word2Vec [9], has achieved promising results in the area of natural language processing, such as syntactic parsing [10, 11], named entity recognition [12], sentiment analysis [13], and WSI [5, 6, 14]. Following the successful history of distributional semantic representation in the tasks, we also benefit from this approach to capture the information of the surrounding words in a context to induce word senses.

The structure of this paper is as follows: in Section 2, the distributional semantic representation of words and contexts is described. In Section 3, the previous studies on context clustering for WSI and the clustering algorithms used for this task are discussed. Section 4 explains our proposed models. The experimental results are reported and discussed in Section 5. Finally, Section 6 concludes the paper.

2- Distributional Semantic Representation

Distributional semantics is based upon the "distributional hypothesis". The distributional hypothesis roots at the idea of Harris [15] such that the words that occur in the same context tend to have similar meanings. Harris believes that the meaning of a word is reflected from the context that the word is used. This idea resembles the idea proposed by Wittgenstein [16] who says, "the meaning of words lies in their use". Firth [17] adds that "[y]ou shall know a word by the company it keeps". These ideas indicate that using the contextual information plays a very important role in determining the meaning of a word. As a result, Miller and Charles [16] have proposed a strong contextual hypothesis that expresses "two words are semantically similar to the extent that their contextual representations are similar". Based on their idea, Examples (1) to (3) nicely show that the words '*car*', '*automobile*', and '*auto*' are (relatively) semantically similar due to have (relatively) similar contexts.

(1) He parked his new car in the parking lot.
(2) He parked his new automobile in the parking lot.
(3) He parked his new auto in the parking lot.

To represent the contextual information of the distributional semantics, two general approaches are used [14]: (a) Bayesian methods using topic modeling approaches; and (b) feature-based methods using the vector representation of the contextual information. While topic modeling approaches, such as Latent Dirichlet Allocation [19] and Hierarchical Dirichlet Process [20], represented successful results in WSI, the flexibility of vector space models has received researchers' attention to capture multiple senses of words in the WSI framework.

The vector space model exploited in information retrieval [21] has a crucial contribution to distributional semantics to represent information of a word and its context. In other words, compressing the information about the words and their contexts in vectors explores the semantic distribution of the words. In the literature, this way of encoding and representing word information is known as 'word embedding' [9]. Computing the geometric distance between the vectors results in the similarity between the words. In Examples (1) to (3), the distance between the vectors of the words '*car*', '*automobile*', and '*auto*' is measured low; therefore, these words are assumed to have a similar meaning. There are several similarity measures to compute the vector distance, such as the Euclidean distance, the Cosine similarity, the Jaccard measure, and the Dice measure [22].

Precise coding of the word's contextual information has a direct impact on the quality of finding the most similar words. Since the context plays a very import role, Peirsman and Geeraerts [23] introduced three types of linguistic contexts: (a) document-based model: the words which are used in the same paragraph or in the same documents are similar [24, 25]; (b) syntax-based model: words are compared according to their syntactic relations, more precisely using the dependency relations [26, 27, 28, 10], or the combinatory categorial grammar [29]; and (c) word-based model: words are modeled based on their word-word co-occurrence within a window size. These word co-occurrences resemble the 'bag-of-words' model [25].

In recent studies, the word embedding approach has been taken into the consideration to build the words' vectors. The promising results that this approach obtained caused researchers to propose different techniques to achieve high quality vectors. As a result, two different approaches have been widely studied recently to model the contextual information: (a) using the matrix decomposition techniques, and (b) using the neural network-based techniques. GLObal VEctor representation (GloVe) [30] is an unsupervised learning method that follows the former approach to provide the distributional representation of

words. Continuous Skip gram (Skip-gram) and Continuous Bag Of Words (CBOW) models [9] use the latter approach to represent the contextual information of a word in a vector. Various toolkits are developed based on these approaches, such as the Word2Vec toolkit developed by Mikolov et al. [9]. In this paper, we use the Gensim library in Python [1] to create words' vectors in our model. To capture the context of each word for clustering, we propose two modes within our model. We use the whole sentence and extract the required information of the target word from the sentence, thereafter called the SentContext mode. Additionally, we limit the local context of the target word to the surrounding words and extract the contextual information of the target word with respect to the neighboring words, thereafter called the WinContext mode.

3- Studies on Context Clustering for WSI

The main focus of this paper is on the WSI task that is performed by context clustering to distinguish senses of the target polysemous word. In this approach, each cluster determines a sense of the target word.

Huang et al. [4] calculated TF-IDF[2] of each word and used it as a weighting value in the vectors of each word. The K-means algorithm was used to cluster the weighted words' contexts.

Neelakantan et al. [5] predicted each sense of a word as a context cluster assignment. To this end, they used the K-means algorithm in their model, such that a fixed number of clusters, namely 3 clusters, was defined to run the clustering algorithm.

Li and Jurafsky [6], however, proposed using CRP [7] as a non-parametric model to capture the senses dynamically. In their approach, the model decides either to generate a new sense for each context or to assign the context to an already generated sense.

Wang et al. [31] proposed a model to use weighted topic modeling for sense induction.

Amrami and Goldberg [32] extended a bidirectional recurrent neural network model proposed by Peters et al. [33] and used predicted word probabilities in the language model of their induction model.

Alagic et al. [34] proposed the idea that words belonging to a cluster should be able to be substituted in an appropriate context. Based on this idea, they implemented a model to induce word senses.

Correa and Amancio [35] used the complex network proposed by Contucci et al. [36] for context embedding and proposed a model to capture the structural relationship among contexts.

A large number of researches in this field use context clustering to address the problem. Both parametric and non-parametric methods have been studied in this field. The proposed models by Huang et al. [4], Neelakantan et al. [5], and Amrami and Goldberg [32] are the examples of parametric clustering; while the proposed model by Li and Jurafsky [6] is an example of non-parametric clustering. The main advantage of parametric clustering is that they can work with high data dimensionality; but its main disadvantage, as discussed in Section 1, is requiring a fixed number of clusters, which does not seem to meet the requirements of the WSI task. The advantage of non-parametric methods is that they do not require a fixed number of clusters; but the disadvantage of these methods is their poor performance to make a decision in order to assign a word to a new cluster.

As reported by Song et al. [14], a comparative study on parametric and non-parametric models on the SemEval2010 WSI task [37] shows that the K-means parametric model outperforms the CRP algorithm proposed by Li and Jurafsky [6]. As stated in Song et al. [14], the main reason for obtaining such results is the poor performance of CRP in making a decision to assign a word to a new cluster. While the best average number of clusters in the SemEval2010 gold data for the WSI task for both noun and verb categories is 5.04 clusters (senses), in the study of Neelakantan et al. [5] the K-means algorithm used 3 clusters as the fixed number of clusters and CRP ended to a lesser number of clusters on average. This result indicates that relaxing the pre-defined number of clusters in K-means can further improve the performance of the task.

4- Density Measure as a Clustering Criteria

The K-means algorithm [38] is one of the most popular unsupervised learning algorithms to be used in various tasks. This clustering algorithm works based on the similarity within the objects of a cluster, and the dissimilarities between the objects of different clusters. To make the decision about the similarities, the distance between the objects is approximated. To find the best number of clusters, we need to evaluate the results of the clustering algorithm. The clustering result can be measured externally or internally. In the former validation, gold data is required; while in the latter validation no gold data is required and it is done in an unsupervised fashion. In our case, since we have no access to gold data in the real application runtime, the latter validation has to be used.

Liu et al. [39] introduced five aspects that have impact on selecting appropriate internal validation measures of clustering: (a) the monotonicity of different internal validation indices, (b) the impact of noise, (c) the density

[1] https://radimrehurek.com/gensim/models/word2vec.html
[2] Term Frequency-Inverse Document Frequency

of clusters, (d) the clusters that are closed to each other (sub-clusters), and (e) the skewed distribution of data in clusters. In the WSI task, the most important aspect that should be taken into the consideration for capturing the number of clusters is the 'density'. The noise and the skewed distribution are also relevant aspects for this task, but we do not study them in this paper and leave them for further studies.

The silhouette coefficient score [40], also called silhouette index, is one of the well-known metrics that scales the validity of the clustering result and makes a distinction between the clearly defined clusters and the vague ones. Equation (1) computes the silhouette coefficient score of the instance *i* (*s*(*i*)):

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (1)$$

where *a*(*i*) is the average distance between instance *i* and all instances that are in the same cluster as *i*, and *b*(*i*) is the minimum average distance between instance *i* and all instances in clusters of which *i* is not a member. The value of the silhouette score is between 1 and -1. If the score is closer to 1, it determines that there is high density within the objects of a cluster, and the objects are well clustered. If the score is closer to -1, this indicates that the objects in the cluster are not classified well.

Instead of computing the silhouette score for each individual object of a cluster, the average silhouette score is computed; therefore, the higher this average score, the higher density of the objects in a cluster and well-clustering of the data.

The silhouette coefficient score can provide a way to assess parameters to choose the optimum number of clusters. This property of the score is very relevant and useful for WSI. Using the average silhouette score makes it possible to have a better approximation on the number of clusters; i.e., for each number of clusters, we calculate silhouette and select the cluster number which results the highest value of silhouette. In our research, we use the silhouette coefficient score to measure the density of clusters for the WSI task.

Algorithm (1) shows the pseudo-code of our proposed model. As can be seen in the code, four data sources as the input are required: (a) list of target words to induce their senses; (b) a raw corpus; (c) a collection of unlabeled data containing the target words; and (d) a set of annotated data where the target word is labeled in terms of its sense in the local context. In the first step of our algorithm, Word2Vec is used for representing words in the vector space model. In the next step, the local context of the target words (either at the sentence level or limited to the surrounding words) are extracted. Then, the context vectors of the target words are built from the unlabeled and labeled data based on weighted average word vector.

Algorithm (1): Density-based our proposed WSI model

Input: Target word TW
Raw text corpus *C* for embedding
Set of unannotated data *UD* containing *TW*
Set of annotated data *AD* containing *TW* for evaluation
Step1: Create words' vectors V of concatenated data (*C* + *UD* + *AD*)
Step2: Extract context (either sentence or window) of *TW* from *UD* and *AD*
Step3: Compute TF-IDF of context words to be used as weights
Step4: Create context vector for each instances of *UD* and *AD* based on weighted average word vector
for *i* = 2 **to** *i* = *t* **do**
  Step5: Run K-means clustering algorithm on the created context vectors of *TW*
  Step6: Calculate average silhouette score *Si*
  if *i* > 2 and $\mathbf{S_i < S_{i-1}}$ **then**
    Step7: Stop and select $i-1$ as the optimum cluster number
  end if
end for
if no optimum cluster number is selected **then**
Step8: select *t* as the optimum cluster number
end if

In the next step, the K-means algorithm is run on the created context vectors of a target word. K-means requires a predefined number of clusters. This number varies 2 to 20 and in each experiment the average silhouette score is calculated. As far as the silhouette score is increasing, the number of clusters increases as well. The algorithm is stopped as soon as the silhouette score decreases. In the case of finding no optimum number of clusters, it halts by reaching the upper bound of the loop.

This process results in the time complexity of O(*nkit*), where *n* is the number of instances containing the target word, *k* is the number of clusters, *i* is the number of iterations, and *t* is the number of trials to find the optimum number of clusters which starts from 2 and continues to 20 in the worst case. It should be mentioned that the state-of-the-art models, namely CRP and Kmeans-3, have the time complexity of O(n^2) and O(*nki*), respectively.

5- Experimental Result

5-1- Data Set

To run our experiments, we require three data sets: the labeled data to be used for evaluating the clustering

performance of each target word, the unlabeled data set to be used for clustering each target word, and the data pool for creating the vector representation of the words. To evaluate the clustering results and to create the clusters containing the target words in their contexts, we use the SemEval2010 data set for the WSI task [37] that is mostly from the news domain. In total, 100 words (50 verbs and 50 nouns) are the target words in this data set. This data set contains 8,915 instances as test data with sense annotation and 888,722 unannotated sentences as training data. In the evaluation, two evaluation metrics, namely V-measure and F-measure, are used. These two metrics are explained in Section 5.3.

The data that we use for creating word vectors is The Westbury Lab Wikipedia Corpus developed by Shaoul and Westbury [41]. This corpus that is freely available is collected from the dump of English Wikipedia articles in April 2010. The corpus contains almost 990 million word tokens of the general domain and it has been used for similar tasks as reported in the literature [4, 5]. It should be mentioned that the documents with less than 2000 characters long are excluded from the corpus.

5-2- Baselines

In SemEval2010 [37] three baselines are introduced: (a) the Most Frequent Sense (MFS): in this baseline, all instances are assigned to a single cluster which contains the most frequent sense; (b) one instance per cluster, thereafter named 1S1C: in this baseline, each instance is assigned to a separate cluster; therefore the number of clusters is equal to the number of instances; and (c) random baseline where an instance is randomly assigned to a cluster. The randomization can be done more than once, so that the average result is considered as the final result. In experiments, randomization has been done five times and the average result of the experiments is reported.

Moreover, we use three state-of-the-art models, namely the CRP model proposed by Li and Jurafsky [6], the K-means-3 cluster which assumes 3 senses for each word proposed by Neelakantan et al. [5], and the SemEval2010 Average Participants that is the average system performance of the 26 groups participated in the SemEval2010 WSI task. Furthermore, the state-of-the-art models are considered as additional baselines to compare with the clustering performance of our proposed model.

5-3- Evaluation Metrics

To evaluate the accuracy of the clustering performance, various metrics are proposed. VanRijsbergen [42] proposed F-measure as a metric for evaluating external clustering. Dom [43] and Meila [44] proposed using an entropy-based approach to evaluate how good the clustering result is. Additionally, Rosenberg and Hirschberg [45] proposed V-measure as another entropy-based approach. This metric is a harmonic mean of evaluating both internal and external clustering. Among the metrics in the literature, F-measure and V-measure are frequently used which are explained in more detail.

F-measure proposed by VanRijsbergen [42] is the metric for computing the accuracy of information retrieval as in Equation (2):

$$F - measure = \frac{(1+\beta) \times P \times R}{(\beta \times P) + R} \tag{2}$$

where P is precision, R is recall, and β is a weighting parameter. If $\beta > 1$, more weight is assigned to recall, and in case $\beta < 1$, more weight is assigned to precision. If $\beta = 1$, precision and recall are considered equally. Equations (3) and (4) compute precision and recall, respectively. In all equations, K is the CLUSTER set, which is the hypothesized clusters from the clustering output and C is the CLASS set, which is the correct partitioning of the data; i.e., for a target dataset with N elements, we have two partitions: the guess partition K, and the gold partition C.

$$P = \frac{n_{ij}}{|k_i|} \tag{3}$$

$$R = \frac{n_{ij}}{|c_i|} \tag{4}$$

where n_{ij} is the number of members of class $ci \in C$ that is the element of cluster $kj \in K$.

V-measure computes the harmonic mean of homogeneity, h, and completeness, c, of clustering to capture the clustering success as computed in Equation (5):

$$V - measure = \frac{(1+\beta) \times h \times c}{(\beta \times h) + c} \tag{5}$$

Homogeneity means that in each CLUSTER, there are a few numbers of CLASSes. The best mode of homogeneity is when a cluster consists of only samples of one class. Completeness, which is the reverse of homogeneity, means that each CLASS is appeared in a few numbers of CLUSTERs. The best mode of completeness is when all samples of the same class are within a single cluster.

As Rosenberg and Hirschberg [45] explain, homogeneity and completeness are formally defined as:

(6)

$$h = \begin{cases} 1 & if \quad H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & else \end{cases}$$

where

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{N} log \frac{a_{ck}}{\sum_{c=1}^{|C|} a_{ck}}$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|} a_{ck}}{N} log \frac{\sum_{k=1}^{|K|} a_{ck}}{N}$$

(7)

$$c = \begin{cases} 1 & \text{if } H(K) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{else} \end{cases}$$

where

$$H(K|C) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{N} log \frac{a_{ck}}{\sum_{k=1}^{|K|} a_{ck}}$$

$$H(K) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|} a_{ck}}{N} log \frac{\sum_{c=1}^{|C|} a_{ck}}{N}$$

$C=\{c_i \mid i = 1, \ldots, n\}$ is the set of CLASS, $K = \{k_i \mid 1, \ldots, m\}$ is the set of CLUSTER, and N is the number of data points in the data set, and a_{ck} is the number of elements of class c in cluster k.

The advantage of V-measure over F-measure is that in the evaluation, both homogeneity and completeness are taken into the consideration, while in F-measure, only the distribution of classes in clusters, i.e., homogeneity, is considered and it does not care about whether in each cluster the number of classes are minimized. This difference indicates that V-measure is more reliable than F-measure and it accurately evaluates the performance of the clustering result.

5-4- Setup of Experiments

As mentioned, we use the K-means clustering algorithm for our experiments. This clustering algorithm requires vector representation of the data; consequently, we use the Gensim Python library to create the vectors of the words. The setup of creating the words' vectors and to make the vectors as distinct as possible are as follows: (a) the Skip-gram model is employed for building vectors to better capture the context; (b) to use the Skip-gram model, similar to Huang et al. [4], the information of the local context containing 8 words, 4 words before and 4 words after the target word, is extracted; (c) similar to Neelakantan et al. [5], the dimension of each vector is set to 300; and (d) words with frequency 5 and above are kept to build the vectors. The vector of the rest words is considered zero. In the next step, the context vectors are created as weighted average of words' vectors. Based on the idea proposed by Huang et al. [4], we calculate TF-IDF of each word and use it as a weighting value for each vector to compute the context vector.

In our proposed models, two modes of input data are provided for the K-means algorithm. In the first mode, the 'SentContext' mode, the weighted vectors of the words contained in the sentence are summed up to build the sentence vector of the target word, and then the score is normalized based on the sentence length. In the second mode, the 'WinContext' mode, we use the limited context of the target word, 4 words before and 4 words after the target word, to build the sentence vector. The reason to limit the context in this mode to the 8 words is to be similar to the number of context words used for building the words' vector.

The sketch of our proposed model was described in the pseudo-code of Algorithm (1). In our experiments, the data is clustered with different fixed cluster numbers in the K-means algorithm such that the clusters' number vary from 2 to 20 for both noun and verb categories. We perform clustering for each cluster number, starting from 2. Then for each cluster number, we compute the silhouette coefficient score to measure the density of clusters. As long as the silhouette coefficient score increases, the algorithm adds up to the number of clusters, and the clustering task is reperformed. This task is repeated, and as soon as the silhouette coefficient score is decreased, the clustering process stops, and the optimum cluster number with the highest silhouette score is selected as the best number of clusters.

5-5- Results and Discussion

The performance of our model is evaluated based on the external and internal evaluation methods. In Table (1), the summary of the obtained results of the external evaluation method including the baseline models, the state-of-the-art models, and the two modes of our proposed model are reported.

According to the results, the 1S1C baseline outperformed all of the models according to V-measure and neither modes of our model nor the state-of-the-art models were able to beat this baseline. In contrast, this baseline obtained the lowest score according to F-measure. Moreover, the MFS baseline performed the worst according to V-measure and the best according to F-

measure. These two baselines can be considered as a spectrum such that the MFS baseline (homogeneity) is the worst point and the 1S1C baseline (completeness) is the best point. This indicates that there is a trade-off between the two metrics, and the ultimate goal is to achieve a result that moves towards the reasonable value in both metrics.

Table (1): Results of the baselines, the state-of-the-art models, and our proposed model

	Model	***V-Measure(%)***			***F-measure(%)***		
		all	***noun***	***verb***	***all***	***noun***	***verb***
BASELINE	Random Sampling	4.4	4.2	4.6	31.9	30.4	34.1
	1S1C	**31.70**	35.8	25.6	0.09	0.11	0.08
	MFS	0	0	0	**63.40**	57	72.7
STATE-OF-THE-ART	CRP	5.7	7.4	3.2	**55.3**	49.4	63.8
	Kmeans-3	**9.8**	13.5	4.3	55.1	50.7	61.6
	SemEval2010 Average Participants	6.63	7.08	5.95	8.85	9.44	10.83
PROPOSED MODEL	SentContext	**14.28**	16.6	10.9	**44.59**	42.45	47.7
	WinContext	**15.19**	16.7	13	39.57	36.76	43.66

The random baseline that is closer to the real application has obtained better results than the MFS baseline based on V-measure, and better than the 1S1C baseline based on F-measure. As can be seen in the table, our proposed model and three of the state-of-the-art models have beaten this baseline according to the V- and F-measure for both noun and verb categories.

Based on the reported results, the Kmeans-3 model performed the best among the state-of-the-art models according to V-measure for both categories.

Comparing our proposed model with the baselines for both categories, the two modes of our model outperformed the random sampling model based on both V- and F-measure metrics. We further compared the two modes of our model with themselves. In general, for both categories, the WinContext mode obtained a better performance than the SentContext mode according to V-measure. The performance of the WinContext mode for the verb category is 2.1% better than the SentContext mode; while for the noun category, the performance of the two modes are relatively similar. This achievement determines that considering the local context of verbs can identify the meaning of the word; while for the nouns a wider context might be required which varies from one word to another.

Comparing the two modes of our proposed model with the state-of-the-art models, we observed that our proposed model outperformed the state-of-the-art models according to V-measure for both noun and verb categories, while the F-measure is kept in a reasonable range. Our model has beaten the SemEval2010 Average Participants baseline according to F-measure as well as V-measure for both categories. According to the results, we can conclude that the density within the objects of a cluster has a direct impact on homogeneity and completeness in the V-measure metric; consequently it causes to increase the accuracy of the clustering result. The clustering density property has no impact on the F-measure evaluation metric as seen in the results of the state-of-the-art models that achieved a higher F-measure score than our proposed model.

Additionally, the internal evaluation of the clustering algorithm is performed and the average silhouette score of the selected clusters of the target words is computed. The detailed results of each target word in the gold data for both categories are reported in Table (2).

We further compared the output of our proposed model with the SemEval2010 gold standard data in terms of number of identified senses, reported in Table (3). The average number of clusters and the maximum number of clusters in our proposed model have relatively obtained the expected results in the gold data for both categories and our model captures the number of senses relatively accurate, while in the proposed model by Neelakantan et al. [5] the number of senses is set to 3 which is not accurate compared to the gold data. As seen in this table, the density of clusters in both noun and verb categories of the SentContext mode is higher than the WinContext mode. In addition, the density of the clusters for the verb category in both modes is higher than the density in the noun category. One reason for this is that verbs have smaller number of senses; therefore the clusters are denser than nouns that have larger number of senses.

6- Conclusion

In this paper, we used the K-means clustering algorithm, as a parametric clustering algorithm, for the WSI task. Due to the nature of the parametric clustering algorithm, the number of clusters should be predefined that is not possible for the WSI task. To tackle the problem, we proposed a model that uses the density of the clustering algorithm to identify words' senses. To build the model, word embedding with Skip-gram is utilized for this task. In our experiments, we used the silhouette coefficient score to measure density of clusters and estimate the best number of clusters. The experimental results of the external evaluation metric showed that our proposed model has beaten the state-of-the-art models. The obtained results determined that the high density within the objects of a cluster has a direct impact on well-clustering of objects. Moreover, the average number of clusters and the maximum number of clusters in the output of our proposed model are relatively close to the gold data.

Table 2: Number of clusters of the `NOUN' and `VERB' categories in the Gold Data (*GD*), the output of the SentContext (*SC*) and WinContext (*WC*) modes along with their corresponding Silhouette score (*S*)

NOUN						***VERB***					
Word	***GD***	***SC***	S_{SC}	***WC***	S_{WC}	***Word***	***GD***	***SC***	S_{SC}	***WC***	S_{WC}
access	8	5	0.044	5	0.032	accommodate	3	6	0.067	7	0.033
accounting	5	9	0.062	5	0.037	analyze	2	3	0.074	4	0.047
address	5	6	0.071	7	0.054	appeal	4	7	0.062	7	0.027
air	11	5	0.073	4	0.052	apply	4	6	0.077	6	0.044
body	14	4	0.084	6	0.029	assemble	2	5	0.066	6	0.03
camp	7	6	0.056	8	0.024	assert	3	5	0.071	6	0.04
campaign	4	4	0.064	5	0.039	bow	5	5	0.063	5	0.067
cell	6	3	0.157	5	0.06	cheat	2	4	0.077	3	0.078
challenge	10	3	0.076	3	0.053	commit	3	4	0.091	3	0.089
chip	5	4	0.108	4	0.082	conclude	4	4	0.083	7	0.038
class	6	5	0.041	6	0.024	cultivate	4	6	0.069	4	0.064
commission	8	5	0.063	4	0.039	defend	2	3	0.139	3	0.074
community	7	10	0.041	7	0.025	deny	3	4	0.081	9	0.04
dealer	7	4	0.093	5	0.051	deploy	2	8	0.07	8	0.049
display	5	4	0.075	11	0.022	divide	5	4	0.064	5	0.033
edge	10	3	0.082	6	0.028	expose	2	4	0.072	6	0.049
entry	8	4	0.073	5	0.03	figure	5	6	0.041	7	0.033
failure	7	13	0.077	5	0.056	frame	4	3	0.132	7	0.059
field	6	6	0.071	6	0.048	happen	4	9	0.026	4	0.024
flight	7	4	0.068	9	0.019	haunt	2	9	0.034	5	0.049
foundation	3	8	0.064	7	0.037	insist	2	5	0.062	6	0.02
function	6	3	0.136	6	0.037	introduce	3	5	0.082	6	0.033
gap	7	8	0.057	4	0.05	lay	6	7	0.087	9	0.042
gas	6	6	0.064	5	0.042	level	4	5	0.059	3	0.104
guarantee	10	9	0.052	10	0.032	lie	4	3	0.083	4	0.063
house	13	7	0.051	5	0.038	mount	5	5	0.097	4	0.066
idea	6	7	0.033	10	-0.002	observe	4	4	0.091	7	0.055
innovation	5	12	0.019	4	0.044	operate	2	5	0.05	6	0.032
legislation	4	4	0.04	5	0.024	owe	3	5	0.086	5	0.054
margin	7	7	0.054	7	0.048	pour	4	4	0.156	6	0.066
mark	5	7	0.094	4	0.031	presume	2	7	0.071	4	0.102
market	4	7	0.028	9	0.009	pursue	2	6	0.067	5	0.043
mind	8	5	0.062	6	0.035	question	2	6	0.056	8	0.036
moment	9	6	0.076	7	0.047	reap	2	6	0.095	10	0.129
movement	7	3	0.122	3	0.045	regain	2	4	0.093	4	0.049
note	6	4	0.083	3	0.1	relax	3	3	0.115	4	0.064
office	6	5	0.072	4	0.046	reveal	2	3	0.083	4	0.04
officer	8	3	0.092	10	0.013	root	4	6	0.072	4	0.077
origin	5	4	0.1	6	0.023	separate	2	7	0.086	3	0.072
park	9	6	0.059	7	0.024	shave	2	7	0.055	6	0.059
promotion	5	4	0.084	5	0.057	signal	2	5	0.076	3	0.086
rally	7	6	0.061	5	0.024	slow	2	4	0.106	6	0.019
reputation	11	5	0.089	5	0.047	sniff	3	7	0.063	4	0.089
road	5	4	0.092	4	0.042	stick	4	7	0.042	5	0.056
screen	9	5	0.065	5	0.035	straighten	3	7	0.096	5	0.076
shape	7	4	0.089	6	0.044	swear	5	6	0.054	4	0.067
speed	4	6	0.072	4	0.056	swim	2	5	0.065	3	0.09
television	4	8	0.05	10	0.038	violate	2	4	0.095	4	0.077
threat	8	6	0.027	7	0.026	wait	2	3	0.097	4	0.061
tour	8	6	0.035	8	0.022	weigh	6	5	0.089	5	0.077

Table 3: Comparing the Average Number of Clusters (*ANC*) and Maximum Number of Clusters (*MNC*) along with the Average Silhouette Score (*ASS*) in gold data and our proposed model for both noun and verb categories

Model	***ANC***		***MNC***		***ASS***	
	Noun	***Verb***	***Noun***	***Verb***	***Noun***	***Verb***
SemEval2010 gold-data	6.96	3.12	14	6	-	-
SentContext	5.64	5.22	13	6	0.07	0.078
WinContext	6.03	5.26	11	6	0.038	0.057

References

[1] F. de Saussure, *Cours de linguistique générale*, C. Bally, A. Sechehaye, and A. Riedlinger, Eds. Lausanne, Paris: Payot, 1916.

[2] G. A. Miller, "WordNet: A lexical database for English," *Communications of the ACM*, Vol. 38, No. 11, 1995, pp. 39-41.

[3] E. Hovey, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, "OntoNotes: The 90% solutions," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 57-60.

[4] E. Huang, R. Socher, C. D. Manning, and A. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Jeju Island, Korea: Association for Computational Linguistics, 2012, Vol. 1, pp. 837-882.

[5] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient non-parametric estimation of multiple embeddings per word in vector space," in *Proceedings of the Conference on Empirical Methods in Natural Language.* Doha, Qatar: Association for Computational Linguistics, 2014.

[6] J. Li, and D. Jurafsky, "Do multi-sense embeddings improve natural language understanding?" in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015, pp. 1722-1732.

[7] D. M. Blei, M. I. Jordan, T. L. Griffiths, and J. B. Tenenbaum, "Hierarchical topic models and the nested Chines restaurant process," in *Proceedings of the 16th International Conference on Neural Information Processing Systems*. MIT Press, 2003, pp. 17-24.

[8] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial database with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, E. Simoudis, J. Han, and U. M. Fayyad, Eds. AAAI Press, 1996, pp. 226-231.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z . Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111-3119.

[10] O. Levy, and Y. Goldberg, "Dependency-based word embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Vol. 2. Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 302-308.

[11] I. Beltagy, K. Erk, and R. Mooney, "Semantic Parsing using distributional semantics and probabilistic logic," in *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, Association for Computational Linguistics, 2014, pp. 7-11.

[12] A. Das, D. Ganguly, and U. Garain, "Named entity recognition with word embeddings and Wikipedia categories for a low-resource language," *ACM Transaction on Asian Low-Resource Language Information Processing*, Vol. 16, No. 3, 2017, pp. 1-19.

[13] L. C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings using intensity scores for sentiment analysis," *IEEE/ACM Transaction on Audio, Speech and Language Processing*, Vol. 26, No. 3, 2018, pp. 671-681.

[14] L. Song, Z. Wang, H. Mi, and D. Gildea, "Sense embedding learning for word sense induction," in *Proceedings of the 5th Joint Conference on Lexical and Computational Semantics*. The *SEM 2016 Organizing Committee, 2016, pp. 85-90.

[15] Z. S. Harris, "Distributional structure," *Word*, Vol. 23, No. 10, 1954, pp. 146-162.

[16] L. Wittgenstein, *Philosophical Investigations*, Oxford, UK: Blackwell Publishing Ltd, 1953.

[17] J. R. Firth, "A synopsis of linguistic theory 1930-1955," *Studies in Linguistic Analysis (Special Volume of the Philosophical Society)*, 1957, pp. 1-32.

[18] G. A. Miller, and W. G. Charles, "Contextual correlates of semantic similarity," *Language and Cognitive Processing*, Vol. 6, No. 1, 1991, pp. 1-28.

[19] D. M. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, Vol. 3, 2003, pp. 993-1022.

[20] Y. W. The, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet process," *Journal of the American Statistical Association*, Vol. 101, No. 476, 2006, pp. 1566-1581.

[21] G. M. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, Vol. 18, No. 11, 1975, pp. 613-620.

[22] D. Jurafsky, and J. H. Martin, *Speech and Language Proeccing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2018, https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

[23] Y. Peirsman, and D. Geeraerts, "Predicting strong associations on the basis of corpus data," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 648-656.

[24] T. K. Landauer, and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychological Review*, Vol. 104, No. 2, 1997, pp. 211-240.

[25] M. Sahlgren, *The Word-space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-dimensional Vector Spaces*, Ph.D. dissertation, Stockholm University, Stockholm, Sweden, 2006.

[26] Z. S. Harris, *A Theory of Language and Information: A Mathematical Approach*, Oxford, England: Oxford University Press, 1991.

[27] D. Lin, "Automatic retrieval and clustering of similar words," in *Proceedings of the 17th International Conference on Computational Linguistics*, Morristown, NJ, USA: Association for Computational Linguistics, 1998, Vol. 2, pp. 768-774.

[28] S. Padó, and M. Lapata, "Dependency-based construction of semantic space models," *Computational Linguistics*, Vol. 33, No. 2, 2007, pp. 161-199.

[29] K. M. Hermann, and P. Blunsom, "The role of syntax in vector space models of compositional semantics," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, 2013, Vol. 1, pp. 894-904.

[30] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, Vol. 14, pp. 1532-1543.

[31] J. Wang, M. Bansal, K. Gimpel, B. D. Ziebart, and C. T. Yu, "A sense-topic model for word sense induction with unsupervised data enrichment," *Transaction of the Association for Computational Linguistics*, Vol. 3, 2015, pp. 59-71.

[32] A. Amrami, and Y. Goldberg, "Word sense induction with neural biLM and symmetric pattern," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4860-4867.

[33] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1. New Orleans, Louisiana: Association for Computational Linguistics, 2018, pp. 2227-2237.

[34] D. Alagić, J. Šnajder, and S. Padó, "Leveraging lexical substitutes for unsupervised word sense induction," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, 2018.

[35] E. A. Corrêa, and D.R. Amancio, "Word sense induction using word embeddings and community detection in complex networks," *Physica A: Statistical Mechanics and its Applications*, Vol. 523, 2019, pp. 180-190.

[36] B. Perozzi, R. Al-Rfou', V. Kulkarni, and S. Skiena, "Inducing language networks from continuous space word representations," in *Complex Networks*, P. Contucci, R. Menezes, A. Omicini, and J. Poncela-Casasnovas, Eds. Cham: Springer International Publishing, 2014, pp. 261-273.

[37] S. Manandhar, I P. Klapaftis, D. Dligach, and S. S. Pradhan, "SemEval-2010 task 14: Word sense induction & disambiguation," in *Proceedings of the 5th International Workshop on Semantic Evaluation*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp 63-68.

[38] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. Berkley, California: University of California Press, 1967, Vol. 1, pp. 281-297.

[39] Y. Lie, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *Proceedings of the 2010 IEEE International Conference on Data Mining*. Washington, D.C., USA: IEEE Computer Society, 2010, pp 911-916.

[40] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, Vol. 20, No. 1, 1987, pp 53-65.

[41] C. Shaoul, and C. Westbury, "The Westbury Lab Wikipedia Corpus," 2010, http://www.psych.ualberta.ca/~westburylab/downloads/westburylab.wikicorp.download.html

[42] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. Newton, MA, USA: Butterworth-Heinemann, 1979.

[43] B. E. Dom, "An information-theoretic external cluster-validity measure," IBM, Technical Report, 2001.

[44] M. Melia, "Comparing clusterings – an information based distance," *Journal of Multivariate Analysis*, Vol. 98, No. 5, 2007, pp. 873-8995.

[45]A. Rosenberg, and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic: Association for Computational Linguistics, 2007, pp 410-420.

Masood Ghayoomi received his PhD degree in Computational Linguistics from Berlin Freie University, Berlin, Germany in 2014, and M.S. degree in Computational Linguistics from Nancy2 University, Nancy, France and Saarland University, Saarbrücken, Germany, in 2009. Currently he is a faculty member at the Institute for Humanities and Cultural Studies. His research interests include Computational Linguistics, Natural Language Processing, Machine Learning, Corpus Linguistics, Syntax and Lexical Semantics.

Low-Complexity Iterative Detection for Uplink Multiuser Large-Scale MIMO

Mojtaba Amiri
Department of Electrical Engineering, Shahed University, Tehran, Iran
mojtaba.amr72@gmail.com
Mahmoud Ferdosizade Naeiny*
Department of Electrical Engineering, Shahed University, Tehran, Iran
m.ferdosizade@shahed.ac.ir

Received: 23/Nov/2019 Revised: 20/ Dec /2019 Accepted: 8/Feb/2020

Abstract
In massive Multiple Input Multiple Output (MIMO) or large scale MIMO systems, uplink detection at the Base Station (BS) is a challenging problem due to significant increase of the dimensions in comparison to ordinary MIMO systems. In this letter, a novel iterative method is proposed for detection of the transmitted symbols in uplink multiuser massive MIMO systems. Linear detection algorithms such as minimum-mean-square-error (MMSE) and zero-forcing (ZF), are able to achieve the performance of the near optimal detector, when the number of base station (BS) antennas is enough high. But the complexity of linear detectors in Massive MIMO systems is high due to the necessity of the calculation of the inverse of a large dimension matrix. In this paper, we address the problem of reducing the complexity of the MMSE detector for massive MIMO systems. The proposed method is based on Gram Schmidt algorithm, which improves the convergence speed and also provides better error rate than the alternative methods. It will be shown that the complexity order is reduced from $O({n_t}^3)$ to $O({n_t}^2)$, where n_t is the number of users. The proposed method avoids the direct computation of matrix inversion. Simulation results show that the proposed method improves the convergence speed and also it achieves the performance of MMSE detector with considerable lower computational complexity.

Keywords: Massive MIMO; Iterative Method; Matrix Inversion; Maximum Likelihood; MMSE Detection.

1- Introduction

In the recent years massive multiuser multiple-input and multiple-output (MIMO) or large scale MIMO technology has been suggested for next generation wireless communication systems. In massive MIMO systems, a large number of antennas are used at the base station (BS). This structure makes it possible to detect the transmitted symbols of several users that they are transmitting their symbols at the same time and frequency. Massive MIMO is one of the promising solutions in future wireless communication systems (such as 5G systems) that increases spectral efficiency and it reduces interference [1-6]. In spite of the benefits of massive MIMO systems, there exist several challenges in these systems such as hardware implementation complexity, detection complexity, channel estimation and antenna correlation [7, 8].

The maximum likelihood (ML) detector is the optimal detector which practically is not feasible for a massive MIMO system, due to the exponential increase in computational complexity with increasing number of antennas [9, 10]. Therefore, suboptimal detectors with lower complexities are suggested. To achieve near-optimal performance, the sphere decoding and K-best methods have been suggested for ordinary MIMO systems. But, these methods are not practical for Massive MIMO systems [8, 11, 12].

Linear detection algorithms such as zero-forcing (ZF) and minimum-mean-square-error (MMSE) receivers can achieve a close performance to that of optimal detector in massive MIMO systems, due to the asymptotic orthogonal channel property [2, 13, 14]. Therefore, linear detection algorithms can be employed in massive MIMO systems with a large number of antennas at the BS. Unfortunately, these methods are involved the inversion of a matrix with the size of the number users, which it imposes a considerable computational cost at the receiver side, due to the large number of users [15].

Recently, different algorithms have been proposed to avoid calculating high-dimensional matrix inversion, for example: Gauss-Seidel (GS) [16], Joint Steepest Descent and Jacobi method (JSDJD) [17], Parallelizable Chebyshev Iteration (PCI) [18], Hybrid Iteration Method (HIM) [19], Neumann series (NS) [20-22]. But

* Corresponding Author

performance of these methods is far away from the optimal detector performance [23, 24].

In this paper, a low complexity method has been proposed based on Gram Schmidt method. The proposed method avoids the direct computation of matrix inversion in MMSE detector. The numerical results verify that the proposed detector achieves the near-optimal performance of the MMSE detector with a significantly reduced computational complexity. Performance of the proposed method has been compared with GS, JSDJD, PCI and HIM methods.

The rest of the paper is organized as follows. Section 2 describes the system model of uplink multiuser massive MIMO system. In section 3 the proposed detector and convergence analysis are presented. Complexity analysis is shown in Section 4. In Section 5, the simulation results and discussions about the performance of the proposed algorithm are presented and finally the paper is concluded in Section 6.

Notation: Boldface capital letters and lowercase letters represent matrices and vectors, respectively. $\mathbf{I}_{\mathrm{K}}$ denotes the $\mathrm{K} \times \mathrm{K}$ identity matrix; $(.)^{\mathrm{H}}$, $(.)^{-1}$, $(.)^{\mathrm{T}}$ and $\|.\|$ denotes conjugate transposition, inversion, transposition and the Euclidean norm of a vector, respectively. $\mathbb{C}^{\mathrm{i}\times\mathrm{j}}$ denote the set of all $\mathrm{i} \times \mathrm{j}$ complex matrices.

2- System Model

We consider uplink multiuser massive MIMO system with $\mathrm{n_t}$ single antenna users transmitting data to a base station with $\mathrm{n_r}$ antennas. The transmitted vector $\mathbf{x} = \left[\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_{\mathrm{n_t}}\right]^{\mathrm{T}}$ includes $\mathrm{n_t}$ data symbols that the elements of $\mathbf{x}$ come from the M-QAM constellation with average power σ_{x}^2 per symbol i.e. $\mathrm{E}\{\mathbf{x}\mathbf{x}^{\mathrm{H}}\} = \sigma_{\mathrm{x}}^2\mathbf{I}_{\mathrm{n_t}}$. The received vector at the BS can be represented by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

Where $\mathbf{H} \in \mathbb{C}^{\mathrm{n_r}\times\mathrm{n_t}}$ is the channel matrix between the BS and the $\mathrm{n_t}$ users, whose entries are modeled as independent and identically distributed (iid) complex Gaussian random variables with zero mean and unit variance, and $\mathbf{n}$ is a white Gaussian noise vector with zero mean and correlation matrix of $\mathrm{E}\{\mathbf{n}\mathbf{n}^{\mathrm{H}}\} = \sigma^2\mathbf{I}_{\mathrm{n_r}}$, where σ^2 is the variance of the noise. It is assumed that channel matrix is known perfectly at the BS, but it is unknown at the transmitter.

The BS detects the transmitted symbols, $\mathbf{x}$, knowing the received vector, $\mathbf{y}$ and the channel matrix, $\mathbf{H}$. Since in massive MIMO systems the number of users, n_t, and the number of BS antennas , n_r, may be in order of hundreds, detection methods which are conventionally used in typical (low scale) MIMO systems are not applicable in massive MIMO scenario.

2-1- MMSE Detection

The MMSE detector is a linear detector which minimizes the mean square error between the transmitted vector and its estimation. The MMSE estimation of the vector $\mathbf{x}$ can be denoted by [16]

$$\hat{\mathbf{x}} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H} + \frac{\sigma^2}{\sigma_{\mathrm{x}}^2}\mathbf{I}_{\mathrm{n_t}}\right)^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{y} = \mathbf{A}^{-1}\mathbf{y}_{\mathrm{MF}} \tag{2}$$

Where $\mathbf{y}_{\mathrm{MF}} = \mathbf{H}^{\mathrm{H}}\mathbf{y}$ is the output of matched filter and $\mathbf{A} = \mathbf{H}^{\mathrm{H}}\mathbf{H} + \frac{\sigma^2}{\sigma_{\mathrm{x}}^2}\mathbf{I}_{\mathrm{n_t}}$ is the MMSE filtering matrix**.** It is noteworthy that, the MMSE detection generally is not optimum, but it has been shown that its performance in massive MIMO systems, with a large number of antennas, is very close to optimum Maximum Likelihood (ML) detector [17].

Since in massive MIMO systems, the dimension of matrix $\mathbf{A}_{\mathrm{n_t}\times\mathrm{n_t}}$ is very large, the inversion of this matrix in (2) is very complex. To avoid calculating the inversion of matrix $\mathbf{A}$, which has the complexity of $O({n_t}^3)$, the solution of the following linear equation is found using iterative methods [25].

$$\mathbf{A}\mathbf{x} = \mathbf{y}_{\mathrm{MF}} \tag{3}$$

When $\mathrm{n_r} \to \infty$, $\mathbf{A}$ becomes diagonal dominant, which means $|a_{ii}| > \sum_{j\neq i}^{n_t}|a_{ij}| \quad \forall i$, where a_{ij} is the element of the ith raw and the jth column of the matrix when the matrix $\mathbf{A}$ is diagonal dominant, some iterative methods can be used to solve (3) instead of using the direct matrix inversion.

3- Proposed Detection Algorithm

In this section, an iterative detector based on the Gram Schmidt method is proposed for detection of the transmitted symbols in massive MIMO systems. This method iteratively achieves the near-optimal performance of the MMSE detector without calculation the matrix inversion.

The main advantage of this method is that it converges faster than the previously proposed methods such as NS [20], HIM [19],GS [16] and JSDJD [17]. In [26, 27] the well-known conjugate Gram Schmidt method has been used to solve an approximated solution of linear equations. We have used this method to solve (3) and to achieve estimation of transmitted symbols in massive MIMO systems. The pseudo-code of the proposed algorithm has been shown in Algorithm 1.

Algorithm 1: Proposed Detector

Input: $\mathbf{y}$, $\mathbf{H}$, σ^2 and σ_x^2
parameters: K 'number of iterations'
Output: The final estimation of transmit symbols: $\hat{\mathbf{x}}_{K+1}$

initialization:

1: $\mathbf{A} = \mathbf{H}^H\mathbf{H} + \frac{\sigma^2}{\sigma_x^2}\mathbf{I}_{n_t}$ and $\mathbf{D}$ = diag($\mathbf{A}$)
2: $\hat{\mathbf{x}}_1 = \mathbf{D}^{-1}\mathbf{y}$ 'Initial Estimation'
3: $\mathbf{r}_1 = \mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_1$ and $\mathbf{p}_1 = \mathbf{r}_1$
Iteration:
4: $\mathbf{d}_i = \mathbf{A}\mathbf{p}_i$
5: $a_i = \frac{\mathbf{p}_i^H\mathbf{r}_i}{\mathbf{p}_i^H\mathbf{d}_i}$
6: $\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + a_i\mathbf{p}_i$
7: $\mathbf{r}_{i+1} = \mathbf{r}_i - a_i\mathbf{d}_i$
8: $\mathbf{p}_{i+1} = \mathbf{r}_{i+1} - \sum_{j=1}^{i}\frac{\mathbf{r}_{i+1}^H\mathbf{d}_j}{\mathbf{p}_j^H\mathbf{d}_j}\mathbf{p}_j$
9: if $i > K+1$ break else go back to step 4
10: **return** $\hat{\mathbf{x}}_{K+1}$

The proposed algorithm is initialized using steps 1, 2 and 3. In these steps, the MMSE filtering matrix , $\mathbf{A}$, is calculated. Then in step 2 an estimation of vector $\mathbf{x}$ is calculated using the inversion of $\mathbf{D}$ instead of inversion of $\mathbf{A}$:

$$\hat{\mathbf{x}}_1 = \mathbf{D}^{-1}\mathbf{y} \tag{4}$$

$\mathbf{D}$ is defined as a diagonal matrix which its diagonal elements is equal to the diagonal elements of the matrix $\mathbf{A}$. Sine $\mathbf{D}$ is diagonal the calculation of its inversion in (4) is not complex.

As mentioned before in massive MIMO systems the matrix $\mathbf{A}$ is diagonal dominant, thus $\hat{\mathbf{x}}_1$ is a good estimation for initialization of the algorithm.

In the (i + 1) th iteration, the estimated vector $\hat{\mathbf{x}}_{i+1}$ is calculated by adding a vector in direction of $\mathbf{p}_i$ to the previous estimation $\hat{\mathbf{x}}_i$.

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + a_i\mathbf{p}_i \tag{5}$$

where $\hat{\mathbf{x}}_i$ is the estimated vector after the i th iteration and $\mathbf{p}_i$ is search direction vector. The search directions $\mathbf{p}_i$ is chosen such that the residual error $\mathbf{r}_{i+1}$ is minimized. To find the value of a_i , we use the fact that error in the (i + 1) th iteration should be orthogonal to $\mathbf{p}_i$ ($\mathbf{p}_i^H\mathbf{e}_{i+1} = 0$) [27]. Since the error vector is unknown, in [27], the Gram Schmidt orthogonalization process is used to find the **A**-orthogonal search direction vector $\mathbf{p}_i$, where **A**-orthogonal is defined as follows:

$$\mathbf{p}_i^H\mathbf{A}\mathbf{p}_j = 0, \quad \forall\, j < i \tag{6}$$

The interpretation of (6) is that each direction, after the multiplication by the matrix $\mathbf{A}$ is orthogonal to the direction vectors of the previous iterations.

In [27] it has been shown that the search direction vector $\mathbf{p}_i$ is calculated by

$$\mathbf{p}_i = \mathbf{r}_i - \sum_{j=1}^{i-1}\beta_{ij}\,\mathbf{p}_j \tag{7}$$

which means that $\mathbf{p}_i$ is generated by the subtraction of the previous directions $\mathbf{p}_j$, (j < i) from the residual vector $\mathbf{r}_i$. In equation (7), the coefficients β_{ij} (for j < i) is defined by [27]

$$\beta_{ij} = \frac{\mathbf{r}_i^H\mathbf{A}\mathbf{p}_j}{\mathbf{p}_j^H\mathbf{A}\mathbf{p}_j} \tag{8}$$

The coefficient a_i in (5) is calculated as [27]

$$a_i = \frac{\mathbf{p}_i^H\mathbf{r}_i}{\mathbf{p}_i^H\mathbf{A}\mathbf{p}_i} \tag{9}$$

3-1- Convergence analysis

The convergence of the proposed Gram-Schmit based method depends on the condition number of matrix $\mathbf{A}$ [27]. The matrix $\mathbf{A}$ is Hermitian Positive Definite (HPD), and its condition number is defined as follows:

$$\kappa = \frac{\lambda_{max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})} \tag{10}$$

Where $\lambda_{max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ are the largest and smallest eigenvalues of the matrix $\mathbf{A}$, respectively. Suppose that the exact solution of the linear equation (3) is $\hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{y}_{MF}$, then it has been shown in [27] that

$$\|\hat{\mathbf{x}} - \hat{\mathbf{x}}_{i+1}\| \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i \|\hat{\mathbf{x}} - \hat{\mathbf{x}}_1\| \tag{11}$$

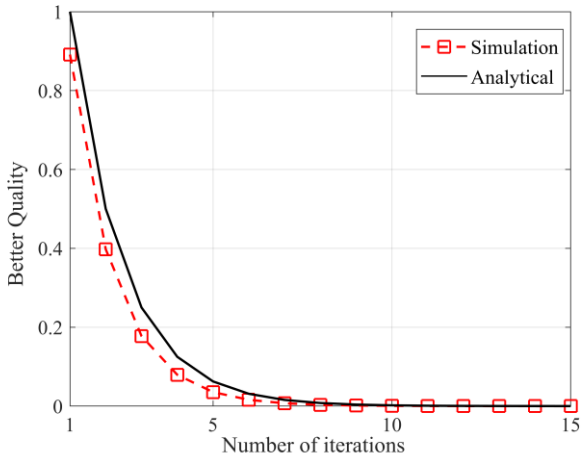

Fig. 1 Comparison between the simulation and analytical estimation error versus the number of iterations with $n_r = 128$ and $n_t = 32$.

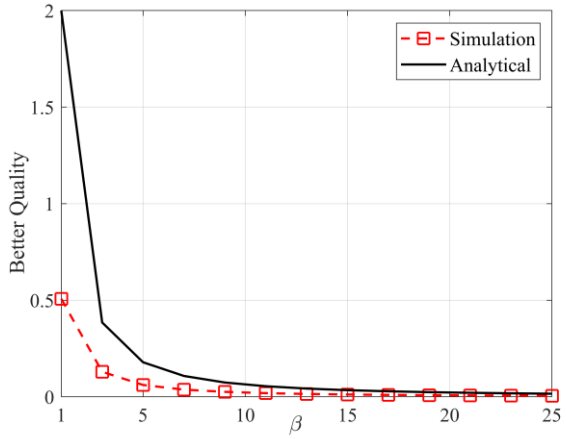

Fig. 2 Comparison between the simulation and analytical estimation error versus $\beta = \frac{n_r}{n_t}$ with $K = 3$ (3 iterations).

Where $\hat{\mathbf{x}}_1$ is the initial estimation obtained by (4). Therefore, after i iterations, the error, $\boldsymbol{e}_{i+1}$ satisfies the following inequality

$$\|\boldsymbol{e}_{i+1}\| \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{i} \|\boldsymbol{e}_1\| \tag{12}$$

If $\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right) < 1$ then based on (12), the error is decreased after each iteration and the algorithm converges.

In massive MIMO systems the largest and smallest eigenvalues of matrix **A** can be approximated by [8]

$$\lambda_{max}(\mathbf{A}) \approx n_t\left(1+\frac{1}{\sqrt{\beta}}\right)^2 \tag{13}$$

and

$$\lambda_{min}(\mathbf{A}) \approx n_t\left(1-\frac{1}{\sqrt{\beta}}\right)^2 \tag{14}$$

Where $\beta = \frac{n_r}{n_t}$.

Using (13) and (14), the estimation error can be obtained as shown in Lemma 1.

Lemma 1. *In large-Scale MIMO systems, the estimation error generated by a detector based on Gram Schmidt method at the i-th iteration can be obtained by*

$$\|\boldsymbol{e}_{i+1}\| \leq 2\left(\frac{n_t}{n_r}\right)^{i/2} \|\boldsymbol{e}_1\| \tag{15}$$

Proof: Substituting the equation (13) and (14) into the equation (10) can be rewritten as follows:

$$\kappa = \frac{\left(1+\sqrt{\beta}\right)^2}{\left(1-\sqrt{\beta}\right)^2} \tag{16}$$

By applying (16) in the equation (12), the inequality (15) is simply derived.

As can be seen from (15), the error is exponentially decreased when i is increased if $n_t < n_r$. The rate of the convergence depends on β. The convergence rate depends on the ratio between the number of users and number of BS antennas.

Fig. 1. shows comparison between the simulation result and analysis of the estimation error versus the number of iterations for $n_r = 128$ and $n_t = 32$. From this figure, it can be seen that the analytical error is very close to simulation error especially when the number of iterations increases. In Fig. 2, the comparison between the simulation and analysis of the estimation error versus β has been repeated, while the number of iterations is assumed to be 3. As can be seen, the distance between the simulation and analytical results is negligible especially when β increases.

4- Complexity Analysis

The computational complexity of this method can be analyzed with respect to the number of multiplications. It has been assumed that the complexity of division operation is the same as the multiplication. In this section, the order of complexity is calculated for proposed method and alternative methods. In the proposed method, we first calculate **A**, $\mathbf{r}_1$ and $\hat{\mathbf{x}}_1$ in steps 1 , 2 and 3 of Algorithm 1 which has the order of complexity of $O({n_t}^2)$ multiplications. The number of required multiplications for K iterations of the Gram Schmidt method is $K({n_t}^2 + 2n_t + K(2n_t + 1)) - n_t$ [27]. Therefore, the total complexity for the proposed algorithm is $O(K{n_t}^2)$. The proposed method requires a similar number of multiplications compared to the JSDJD and GS methods in the same number iterations [16, 17], while the complexity of MMSE method is $O({n_t}^3)$. Thus the proposed algorithm has lower complexity than the MMSE algorithm. Since n_t is usually large for large-scale MIMO systems, it can be observed that the proposed algorithm can evidently reduce the complexity, which it means that the proposed algorithm is suitable for large-scale MIMO systems. As mentioned before, the complexity order of the proposed method and other methods is $O(K{n_t}^2)$, but as it will be shown in simulations the proposed method converges in lower number of iterations than that of other methods.

5- Simulation Result

In this section, performance of the proposed detector in massive MIMO system with 64-QAM and 16-QAM modulations have been evaluated. It is also assumed in all detection methods that the receiver knows the channel matrix completely.

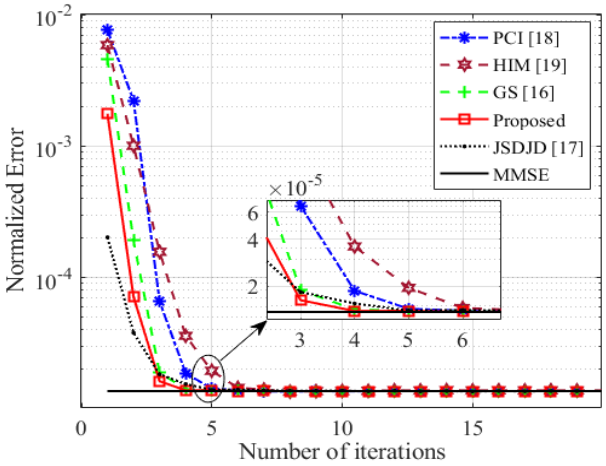

Fig. 3 Normalized estimation error versus the number of iterations for different algorithms at SNR = 15 dB with 64-QAM modulation for $n_r = 128$ and $n_t = 16$.

Fig.3. shows error of different detection methods versus number of iterations for 64-QAM modulation, when SNR is 15dB with $n_r = 128$ and $n_t = 16$. The normalized error has been defined by

$$NE = \frac{\|\mathbf{x}^* - \mathbf{x}\|}{\|\mathbf{x}\|} \quad (17)$$

Where $\|\mathbf{x}^*\|$ is the output of each method and $\mathbf{x}$ is the exact vector of transmitted symbols.

As can be seen, since the number of BS antennas is more than the number of users, all methods converge to the estimation error of the original MMSE detector in a few number of iterations.

In Fig. 4. simulations have been repeated for $n_r = 128$ and $n_t = 64$. As it can be seen form this figure, when the number of users increases, the JSDJD algorithm can not converge and performances of PCI method and HIM method are degraded. The PCI and HIM converge to the original MMSE detector after 10 and 18 iterations, respectively while our proposed and GS converge faster than PCI and HIM methods.

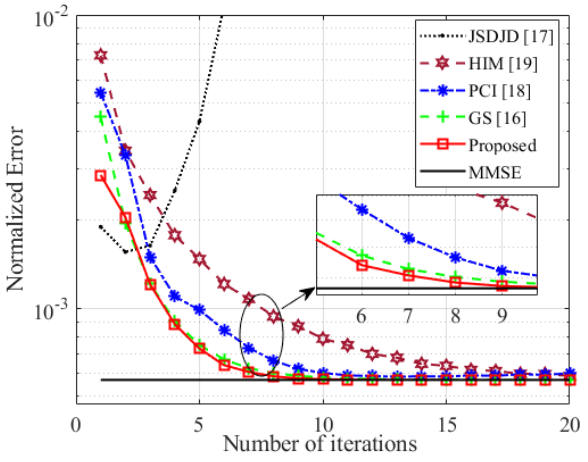

Fig. 4 Normalized estimation error versus the number of iterations for different algorithms at SNR = 15 dB with 64-QAM modulation for $n_r = 128$ and $n_t = 64$.

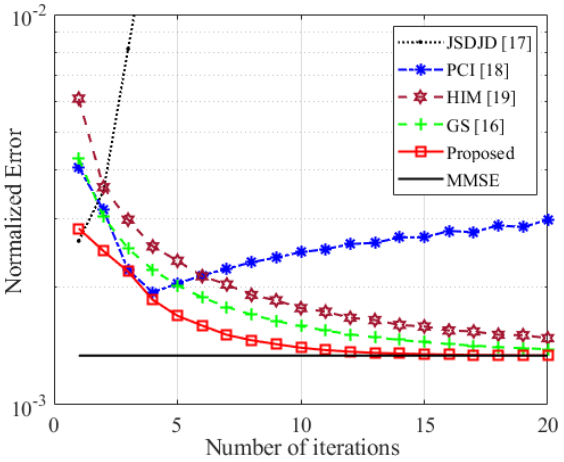

Fig. 5 Normalized estimation error versus the number of iterations for different algorithms at SNR = 15 dB with 64-QAM modulation for $n_r = 128$ and $n_t = 128$.

Fig. 5. Shows error of different detection methods versus number of iterations with $n_r = 128$ and $n_t = 128$. In this case, the JSDJD and PCI algorithms do not converge and performance of Gauss-Seidel and HIM methods are not enough close to that of MMSE detector after even 20 iterations, while performance of proposed method is very close to the MMSE detector after only 10 iterations.

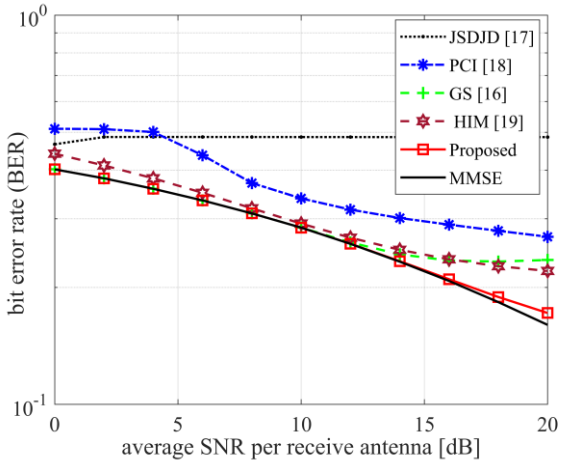

Fig. 6 BER performance comparison between the proposed and other methods in the uplink massive MMO for 64-QAM modulation for $n_r = 128$ and $n_t = 128$ with $K = 15$ (15 iterations).

In summary, performance of the Gauss-Seidel, JSDJD, HIM and PCI methods are degraded when the number of users becomes comparable with the number of BS antennas. Even JSDJD and PCI methods do not converge, when the number of users is close to the number of BS antennas. Unlike these methods, the normalized error of the proposed algorithm converges to that of MMSE detector after a few number of iterations, even when the number of users is close to the number of BS antennas.

Fig. 6 shows BER of different detection methods versus SNR for 64-QAM modulation. In this figure, the number of base station antennas and the number of single-antenna users are considered to be $n_r = 128$ and $n_t = 128$. As it can be seen form this figure, the proposed method has a very close BER to that of original MMSE detector while the performances of Gauss-Seidel, HIM and PCI methods are not enough close to that of MMSE detector. It should be noted that, the JSDJD method does not converge at all and its BER is 1/2. As can be seen, the proposed method has about 5dB and 6dB performance improvement compared with HIM and Gauss–Seidel algorithms, respectively. In the following, all simulations have been repeated for 16-QAM modulation.

Fig.7. shows normalized error of different detection methods versus number of iterations for 16-QAM modulation when SNR is 15dB with $n_r = 128$ and $n_t = 64$. As it can be seen form this figure, the JSDJD algorithm can not converge to the estimation error of the original MMSE detector.

The HIM and Gauss-Seidel methods converge to the original MMSE detector after 17 and 11 iterations, respectively. Also, performance of PCI method is not enough close to that of MMSE detector after 20 iterations, while performance of proposed method is very close to the MMSE detector after only 9 iterations.

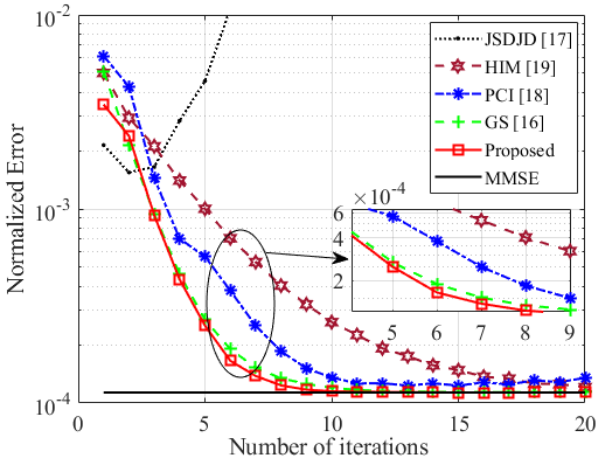

Fig. 7 Normalized estimation error versus the number of iterations for different algorithms at SNR = 15 dB with 16-QAM modulation for $n_r = 128$ and $n_t = 64$.

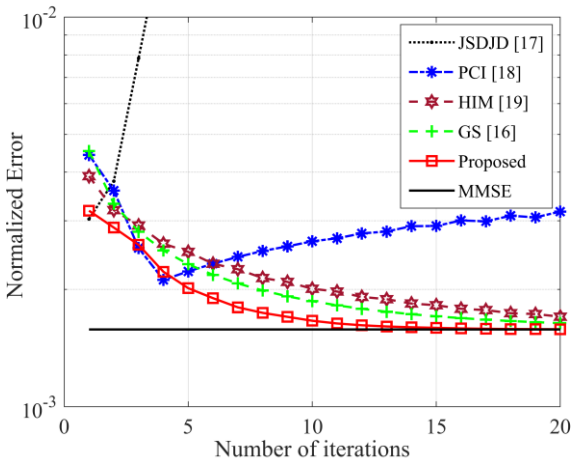

Fig. 8 Normalized estimation error versus the number of iterations for different algorithms at SNR = 15 dB with 16-QAM modulation for $n_r = 128$ and $n_t = 128$.

In Fig. 8 simulations have been repeated for $n_r = 128$ and $n_t = 128$. In this case, the JSDJD and PCI algorithms do not converge to the normalized error of the original MMSE detector and performance of Gauss-Seidel method is not enough close to that of MMSE detector after 20 iterations, while performance of proposed method is very close to the MMSE detector after only 10 iterations. Also, performance of HIM method is not enough close to that of MMSE detector even after 20 iterations.

As it can be seen form this figure, when the number of users increases, the JSDJD algorithm can not converge to the estimation error of the original MMSE detector and performance of PCI method and HIM method are degraded. The PCI and HIM converge to the original MMSE detector after 10 and 18 iterations, respectively.

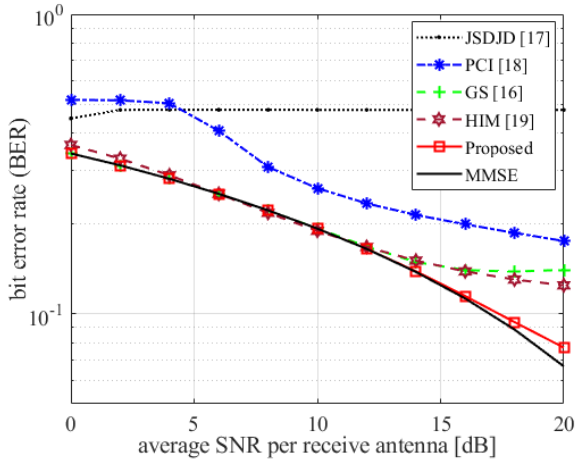

Fig. 9 BER performance comparison between the proposed and other methods in the uplink massive MMO for 16-QAM modulation for $n_r = 128$ and $n_t = 128$ with $K = 15$ (15 iterations).

Fig. 9 shows BER of different detection methods versus SNR for 16-QAM modulation. In this figure, the number of base station antennas and the number of single-antenna users are considered to be $n_r = 128$ and $n_t = 128$. As it can be seen form this figure, the proposed method is able to converge to the performance of the original MMSE detector, while the performances of Gauss-Seidel, HIM and PCI methods are not enough close to that of MMSE detector. It should be noted that, the JSDJD method does not converge and its BER is 1/2. As can be seen, the proposed method has about 5dB and 6dB performance improvement compared with HIM and Gauss–Seidel algorithms, respectively.

It was demonstrated that the proposed method always converges to the original MMSE detector even when the number of users is very close to the number of BS antennas. But the propose method has a disadvantage. The main disadvantage is that when the number of users is close to the number of BS antennas (it is sometimes called loaded scenario), the proposed algorithm needs a high number of iterations to converge. This leads to the increase of the complexity. In future works, we try to modify the proposed method to accelerate its convergence in loaded scenario.

6- Conclusions

In this paper, a novel low complexity iterative detection algorithm for multiuser massive MIMO uplink detection without complicated matrix inversion was proposed. It was shown that the proposed method always converges to the original MMSE detector even when the number of users is very close to the number of BS antennas and its performance is very close to the original near optimum MMSE detector.

References

[1] J. Thompson, X. Ge, H.-C. Wu, R. Irmer, H. Jiang, G. Fettweis, et al., "5G wireless communication systems: Prospects and challenges," IEEE Communications Magazine, vol. 52, pp. 62-64, 2014.

[2] E. Björnson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: Ten myths and one critical question," IEEE Communications Magazine, vol. 54, pp. 114-123, 2016.

[3] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive MIMO detection techniques: a survey," IEEE Communications Surveys & Tutorials, vol. 21, pp. 3109-3132, 2019.

[4] P. Zhang, X. Yang, J. Chen, and Y. Huang, "A survey of testing for 5G: Solutions, opportunities, and challenges," China Communications, vol. 16, pp. 69-85, 2019.

[5] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, et al., "What will 5G be?," IEEE Journal on Selected areas in Communications, vol. 32, pp. 1065-1082, 2014.

[6] Y. Arjoune and S. Faruque, "Artificial Intelligence for 5G Wireless Systems: Opportunities, Challenges, and Future Research Direction," in 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, pp. 1023-1028.

[7] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," IEEE Communications Magazine, vol. 52, pp. 186-195, 2014.

[8] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, et al., "Scaling up MIMO: Opportunities and challenges with very large arrays," IEEE Signal Processing Magazine, vol. 30, pp. 40-60, 2013.

[9] M. O. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," IEEE Transactions on Information Theory, vol. 49, pp. 2389-2402, 2003.

[10] C. Thrampoulidis, W. Xu, and B. Hassibi, "Symbol error rate performance of box-relaxation decoders in massive MIMO," IEEE Transactions on Signal Processing, vol. 66, pp. 3377-3392, 2018.

[11] E. G. Larsson and J. Jalden, "Fixed-complexity soft MIMO detection via partial marginalization," IEEE transactions on Signal Processing, vol. 56, pp. 3397-3407, 2008.

[12] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," IEEE Journal on Selected areas in Communications, vol. 24, pp. 491-503, 2006.

[13] N. A. Khan, S. Ahmed, D. M. S. Bhatti, M. Z. Tunio, and S. Kim, "Study of MIMO detection schemes for emerging wireless communications," International Journal of Computer Science and Network Security, vol. 18, pp. 142-149, 2018.

[14] Y. N. Ahmed, "Enhanced low-complexity matrix inversion method for Massive MIMO systems," in 2019 16th International Symposium on Wireless Communication Systems (ISWCS), 2019, pp. 495-499.

[15] C. Tang, Y. Tao, Y. Chen, C. Liu, L. Yuan, and Z. Xing, "Approximate iteration detection and precoding in Massive MIMO," China Communications, vol. 15, pp. 183-196, 2018.

[16] L. Dai, X. Gao, X. Su, S. Han, I. Chih-Lin, and Z. Wang, "Low-complexity soft-output signal detection based on Gauss–Seidel method for uplink multiuser large-scale MIMO systems," IEEE Transactions on Vehicular Technology, vol. 64, pp. 4839-4845, 2015.

[17] X. Qin, Z. Yan, and G. He, "A near-optimal detection scheme based on joint steepest descent and Jacobi method for uplink Massive MIMO systems," IEEE Communications Letters, vol. 20, pp. 276-279, 2016.

[18] G. Peng, L. Liu, P. Zhang, S. Yin, and S. Wei, "Low-computing-load, high-parallelism detection method based on Chebyshev iteration for Massive MIMO systems with VLSI architecture," IEEE Transactions on Signal Processing, vol. 65, pp. 3775-3788, 2017.

[19] F. Jin, Q. Liu, H. Liu, and P. Wu, "A Low Complexity Signal Detection Scheme Based on Improved Newton

Iteration for Massive MIMO Systems," IEEE Communications Letters, vol. 23, pp. 748-751, 2019.
[20] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations," IEEE Journal of Selected Topics in Signal Processing, vol. 8, pp. 916-929, 2014.
[21] D. Zhu, B. Li, and P. Liang, "On the matrix inversion approximation based on Neumann series in massive MIMO systems," in 2015 IEEE International Conference on Communications (ICC), 2015, pp. 1763-1769.
[22] X. Liu, Z. Zhang, X. Wang, J. Lian, and X. Dai, "A Low complexity high performance weighted Neumann series-based Massive MIMO detection," in 2019 28th Wireless and Optical Communications Conference (WOCC), 2019, pp. 1-5.
[23] M. A. Albreem, "Approximate matrix inversion methods for massive mimo detectors," in 2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT), 2019, pp. 87-92.
[24] M. A. Albreem, A. A. El-Saleh, and M. Juntti, "Linear Massive MIMO uplink detector based On joint jacobi and Gauss-Seidel methods," in 2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020, 2020, pp. 1-4.
[25] M. Burger, B. Kaltenbacher, and A. Neubauer, "Iterative solution methods," in Handbook of Mathematical Methods in Imaging, ed: Springer, 2011, pp. 345-384.
[26] R. I. Hernández, R. Baghaie, and K. Kettunen, "Implementation of Gram-Schmidt conjugate direction and conjugate gradient algorithms," in Proc. IEEE Finish Signal Processing Symposium, 1999, pp. 165-169.
[27] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," ed: Carnegie-Mellon University. Department of Computer Science, 1994.

Mojtaba Amiri received the B.Sc. degree in electrical engineering from Shiraz University of Technology, Iran, in 2015. He received the M.Sc. degree in electrical engineering from Shahed University, Iran, in 2018. His research interests include signal detection in large-scale MIMO systems, direction of arrival (DoA) estimation, and compressive sensing (CS).

Mahmoud Ferdosizade Naeiny was born in Iran, in 1979. He received the B.Sc. degree (Hons.) from the Amirkabir University of Technology (Tehran Polytechnic) in 2002 and the M.Sc. and Ph.D. degrees from the Sharif University of Technology in 2004 and 2010, respectively. He is currently an associate professor with Shahed University. His research interests include array signal processing, compressive sensing, design and implementation of OFDM systems including the synchronization issues, software-defined radio, MIMO systems and the application of signal processing techniques in communications. He has been the Chair **with** the Institute of Modern Technologies on Information and Communications, Shahed University, since 2014**.**

Effective Query Recommendation with Medoid-based Clustering using a Combination of Query, Click and Result Features

Elham Esmaeeli-Gohari
Faculty of Computer Engineering, Yazd University, Yazd, Iran
elhamesmaeeli@stu.yazd.ac.ir
Sajjad Zarifzadeh*
Faculty of Computer Engineering, Yazd University, Yazd, Iran
szarifzadeh@yazd.ac.ir

Received: 20/Nov/2019 Revised: 21/Dec/2019 Accepted: 29/Feb/2020

Abstract
Query recommendation is now an inseparable part of web search engines. The goal of query recommendation is to help users find their intended information by suggesting similar queries that better reflect their information needs. The existing approaches often consider the similarity between queries from one aspect (e.g., similarity with respect to query text or search result) and do not take into account different lexical, syntactic and semantic templates exist in relevant queries. In this paper, we propose a novel query recommendation method that uses a comprehensive set of features to find similar queries. We combine query text and search result features with bipartite graph modeling of user clicks to measure the similarity between queries. Our method is composed of two separate offline (training) and online (test) phases. In the offline phase, it employs an efficient *k*-medoids algorithm to cluster queries with a tolerable processing and memory overhead. In the online phase, we devise a randomized nearest neighbor algorithm for identifying most similar queries with a low response-time. Our evaluation results on two separate datasets from AOL and Parsijoo search engines show the superiority of the proposed method in improving the precision of query recommendation, e.g., by more than 20% in terms of p@10, compared with some well-known algorithms.

Keywords: Recommendation Systems; Search Engine; Clustering; Query; Click.

1- Introduction

Nowadays, search engines play a critical role in giving users access to their needed information over the Internet. The user starts the search process via submitting a query to the search engine. The search engine processes the query and returns a result page containing an ordered list of URLs. The user reviews the excerpt of each URL and chooses some of them for further examination. If the user is satisfied with the returned information, the search process will be finished; otherwise, he/she submits another query. This process may include several cycles from submitting query to browsing results to be finished [1].
One of the most important challenges of search engines is to reduce the response time of queries, while returning the most relevant results. Both issues have a straight impact on users' satisfaction and also the efficiency of search engines. Most of the time, it is hard for users to precisely and clearly express their needs through submitted queries [2]. Therefore, search engines should smartly recommend queries that could lead users faster to the information they want, while keeping the semantic relevant to the original, submitted query [3, 4, 5]. In such cases, it is helpful to use the historical data collected from interactions of users with the search engine (e.g., past queries and clicks) [6].
There are several factors that make the problem of identifying users' intention from search challenging for information retrieval systems. One factor is the existence of polysemous words in queries [7]. For example, the term "jaguar" could refer either to a feline from the same genus as cheetahs, or to the vehicle brand of Jaguar car. There are hundreds of similar examples in other languages as well. For example, the term "شیر" in Persian means milk, lion, and faucet. On the other hand, there are many words that are synonym (e.g., crash, accident, and collision) and different users may use different queries for the exactly same topic.
In the past 15 years, three major approaches have been proposed for query recommendation. In the most popular approach, queries are clustered based on their similarity (e.g., with respect to content or result) and then most similar queries are recommended from closest clusters. There is also another approach that constructs a profile for each user via storing his/her past interactions with the search engine and then recommends queries according to the constructed profile. The final approach employs graph modeling in which a graph structure is built between

* Corresponding Author

queries using either subsequent queries in a session or common URL clicks made after queries. This graph is then used to find most similar queries.

The current approaches usually define the notion of similarity between queries from one aspect (e.g., text, result or click similarity), while neglecting other important features shared between similar queries. This can negatively affect the precision of query recommendation. In addition, the high processing and memory overhead of running recommendation algorithms is prohibitive in many cases. The aim of this research is to overcome the shortcoming of existing algorithms while taking their advantages, such that a high precision is attained while preserving the system efficiency in terms of CPU and RAM usages.

In this paper, we propose a new query recommendation method that combines the benefits of two well-known approaches, i.e., query clustering and graph modeling, together. Our method is composed of two separate offline and online phases. In the offline phase, we cluster our training queries based on three kinds of features, i.e., query n-grams, bipartite query-click graph and search results. The combination of these features can accurately identify relevant queries. Then, we use an efficient version of *k*-medoids algorithm to cluster queries. In the online phase, a fast randomized algorithm is proposed to approximate the most similar queries from the closest cluster. Our evaluations on the search log of two real search engines, Parsijoo (the first Persian web search engine) and AOL [8], show that the proposed method can achieve at least 7% better precision and 23% higher p@10 in comparison with three well-known query recommendation algorithms. Moreover, it incurs a low response time with an affordable memory overhead.

The rest of the paper is organized as follows: In the next section, we review the related works. Section 3 is devoted to the explanation of the proposed method. The results of our evaluations are presented in Section 4. Finally, we conclude the paper in Section 5.

2- Related Works

Query recommendation can be done in different forms, from spelling correction to query auto-completion and query expansion. As an early attempt, Zhang et al. propose to model users' sequential search behavior for recommending queries [9]. Since then, different techniques have been suggested for query recommendation which can be generally divided into query clustering [10, 11, 12, 13], constructing user profile [14, 15, 16] and graph modeling [17-22] categories.

In clustering-based techniques, queries are clustered and then recommended according to their similarity with respect to various factors. For example, Baeza-Yates et al., represent each query with a term-weight vector [11]. Their idea is that queries that have the same meaning may not have common terms, but they may have similar terms in documents selected by users. The term-weight vector of each query is calculated by aggregating the term-weight vectors of documents clicked as a result of query. Each term is weighted based on the number of its occurrences in the clicked documents as well as the number of clicks on documents containing that term. After that, clustering methods are used to find out similar queries. Chaudhary et al., propose to cluster queries based on query content and also the history of users' clicks [13]. In their paper, two principles are used: 1) if two submitted queries contain the same or similar terms, they probably convey the same information needs and therefore they can be located in the same cluster; 2) if two submitted queries lead to clicks on the same URLs, they are similar.

Some other works on query recommendation construct a profile for each user which includes all important interactions of user with the search engine (e.g., submitted queries and clicked URLs). The main idea is to use the users' search history to better understand their intention of search and resolve ambiguities for recommending queries [15]. In early works, there were three ways to construct user profiles: (1) utilizing user relevance feedback: the user is asked to specify whether the visited documents are relevant to his/her needs or not. If the document is relevant, it will be used to identify user's real needs [14]; (2) utilizing user preference: at the time of signing up, the user is asked to submit his preferences and personal information, such as interests, age and education background; (3) utilizing user ranking: the user is asked to rank each of the specified documents from 5 (very bad) to 1 (very good), based on its relevance to his information needs.

The above methods are costly from the viewpoint of users; they often prefer easier and faster methods. In [15], the user profile is constructed using the frequency of each term in user's visited documents. In other words, a user profile consists of a set of vectors where each vector represents a user session in which keywords of the documents clicked during the session along with their frequencies in these documents are stored. The aggregation of these vectors is then used for query recommendation. Since the current session denotes users' recent preferences, the weight of current session is set to a larger value than past sessions. Recently, in [16], the authors propose to construct user profiles for re-ranking search results. They first classify user's clicked URLs into hierarchical categories and then constructs user's profile with respect to these categories.

The third approach for query recommendation is based on graph modeling. Zhang et al. introduce a graph structure where vertices indicate queries and edges between vertices represent the textual similarity between corresponding

queries [9]. An edge is weighted by a damping factor, which denotes the similarity of two consecutive queries in the same session. The similarity of two queries that are not neighbors is obtained by multiplying the weight of edges over the path joining them. In [17], the "Query-URL" bipartite graph is used for query recommendation. In this graph, vertices are queries and URLs, and each edge $e = (q_i, u_j)$ exists in the graph, if and only if, URL u_j is clicked by a user as a result of query q_i. The weight of edge e denotes the number of times users have access to URL u_j via query q_i. After constructing the graph, the similarity between queries is computed using a random walk technique.

Another graph modeling method is to use query-flow graph within sessions [18]. The idea is to utilize the sequence of queries and the modifications that users make to their queries until they reach their intended results. The query-flow graph is constructed based on queries that users submit in different sessions. Here, vertices are distinct queries and each edge (q_i, q_j) indicates that at least in one session query q_j has been submitted right after query q_i. The weight of edge (q_i, q_j) is assigned based on two factors: 1) the probability that two queries q_i and q_j belong to the same search mission and 2) the relative frequency of (q_i, q_j) pair and query q_i. The drawback of this method is that almost half of query pairs occur only once in users' searches, and therefore this graph is sparse. On the other hand, the query-flow graph is asymmetric, because more than 90% of the edges are not reciprocal. Hence, it is not certain that two neighbor queries are equivalent [18].

In order to improve the query-flow graph, a framework is proposed in [19], in which users' querying behaviors are modeled as a Markov-chain process. Furthermore, Bai et al. propose a new approach using an intent-biased random walk algorithm to reduce the sparsity problem in query-flow graphs [20]. The main disadvantage of two latter works is that their accuracy is fairly low, since they do not use click information during graph construction.

Some recent works have adopted different approaches for query recommendation. In [23], the authors propose a context-aware query recommendation which uses a sequence of sequence models (seq2seq) along with a version of Hierarchical Neural Networks for encoding submitted queries in a session and producing the best possible sequence of terms as the next query. This approach considers each query merely as a sequence of terms and hence, it takes into account neither click nor result level information.

In [24], a Knowledge Base (KB) technique is introduced to generate query recommendations based on named entities existing in queries. In order to improve the precision for short queries or queries that their entities are not identifiable, two hybrid methods (named as KB-QREC and D-QREC) are suggested in [25] where KBs and click information are used to retrieve entity relationship information.

To devise a practical recommendation method, we should make a trade-off between the precision and the computational performance of recommendation algorithm. In this paper, we suggest to use a complete set of features along with efficient clustering and nearest neighbor algorithms to improve both parameters at the same time.

3- The Proposed Method

The general workflow of our method is illustrated in Fig. 1. Our method comprises two separate offline and online phases (or equivalently, train and test phases). In the offline phase, we preprocess our training data, obtained from data logs of search engines, via performing spell checking, writing unification, stemming, synonym labeling and stop word removal tasks. Then, we cluster training queries using three kinds of extracted features: "N-grams feature", "Bipartite graph feature" and "Top-k search results and their ranks feature". We employ a simple and fast *k*-medoids algorithm for query clustering. In the online phase, upon the arrival of a test query, we first preprocess the query (just as the offline phase) and then find similar queries through identifying the closest cluster to that query and finally approximating the most similar queries inside that cluster.

3-1- Data Preprocessing Step

In the preprocessing step (which is common in both offline and online phases), the writing structure of queries is unified as much as possible. This step involves the following tasks:

Spell checking: the spelling errors in queries are corrected. For example, the query "TOFL exam" is modified to "TOEFL exam" or the Persian query "Tarneh Alidoosi" [as an Iranian actress] is modified to "Taraneh Alidoosti".

Writing unification: the punctuation marks (? : ! . - ,) are removed, since these marks result in different forms of writing a word. On the other hand, some of the letters in Persian and Arabic alphabets have different shapes, but are used interchangeably (for example, ی and ئ). Therefore, such letters are normalized and replaced with the same Unicode. Furthermore, plural signs are removed. For example, the query "programs" is modified to "program". Also, all English letters are converted to lower case. For instance, both queries "JAVA" and "Java" are converted to "java".

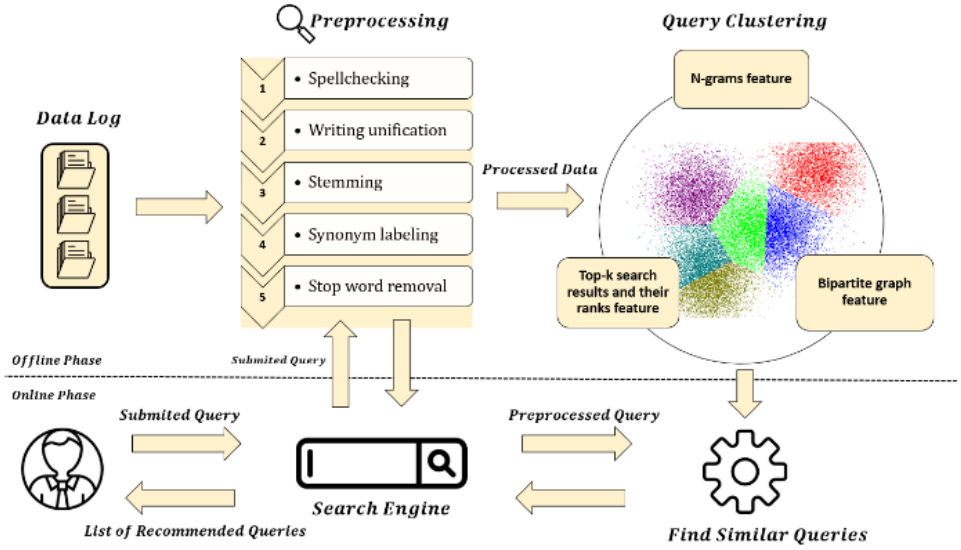

Fig. 1: Workflow of the proposed method

Stemming: All words are replaced with their stems. In this way, the diversity of words in queries is reduced which increases the chance of detecting similar queries. For example, if a user submits a query containing the word "analyze", then queries containing the words "analysis", "analyzing", "analyzer", "analyzes" and "analyzed" could also be related to the user's search needs. Previous studies have shown that stemming reduces the size of bag-of-words by about 50% [26].

Synonym labeling: All words with the same meaning are labeled with the same label such that the distance between similar queries are reduced. As a result of this task, each word is replaced with its most frequently used synonym. For example, the words "battle", "clash", "attack" and "fight" are synonyms and therefore the word "fight" is used as their label.

Stop word removal: The stop words are identified and removed from the submitted queries by using a fixed list of stop words. The stop words are words that appear frequently in queries and documents, but do not imply any particular meaning. Such words consist of articles, prepositions, conjunctions, pronouns and some specific nouns and verbs. As an example, terms like "however" are removed from queries. This task improves both the precision and the computational performance of recommendation system.

These tasks are all performed, using open-source tools available for Persian and English languages, i.e., Hunspell package [27] and PersianStemmer [28].

3-2- Query Clustering Step

Due to its simplicity and performance gains, query clustering has attracted the attention of many researchers in the past years [12, 13]. After accomplishing the data preprocessing step, we cluster our training queries. We argued earlier that terms cannot solely represent the purpose of a query and as a result, it is not precise to just consider the terms of queries to measure their similarity. In this paper, we use various lexical and semantic features to find out similar queries. After the data preprocessing phase, training queries are clustered. In order to compute the similarity between queries, the following three features are extracted:

- N-grams feature
- Bipartite graph (bigraph) feature
- Top-k search results feature

3-2-1- N-grams Feature

If two queries have the same words, they are supposed to indicate similar information needs. To obtain the lexical similarity between queries, we extract the N-grams (up to trigram) of each query. For example, for the sample query "find research council site", we have:

- Unigrams: "find", "research", "council", "site";
- Bigrams: "find research", "research council", "council site";
- Trigrams: "find research council", "research council site".

We construct an N-gram vector for each query. Our vocabulary is the set of all different 1/2/3-grams in submitted queries. An N-gram g exists in the vector of query q, if g appears in q at least once. Similar to the previous works such as [11], the N-gram vector is then

used to obtain the similarity measure between queries. For instance, the vector for the above example includes the following N-grams:
{find, research, council, sites, find research, research council, council sites, find research council, research council sites}
We use the simple Jaccard similarity [35] to compute the similarity between two queries q_x and q_y:

$$Sim_{ngrams}(q_x, q_y) = \frac{|Ngrams(q_x) \cap Ngrams(q_y)|}{|Ngrams(q_x) \cup Ngrams(q_y)|} \quad (1)$$

where $Ngrams(q)$ represents the set of N-grams exist in the vector of query q. In other words, the above similarity indicates the ratio of common N-grams between two queries. The average number of terms in queries is usually limited to a small value, e.g., around 4 in our dataset. Thus, it is easy to compute the above formula. The obtained similarity from the N-grams feature would be more reliable for long queries and may fail for short queries. For example, consider two queries "Apple pear" as a fruit, and "Apple Inc." as a company. According to this feature, the similarity between two queries is 20%, despite the fact that they convey different meanings. On the other hand, many queries (like "natural disasters" and "unforeseen events") denote the same purpose, but they are expressed with different terms. We need extra features to unravel such problems.

3-2-2- Bipartite Graph (Bigraph) Feature

The second feature is based on this assumption that queries that lead to a click on the same documents are likely to represent the same information needs [13]. For this reason, the "Query-URL" bipartite graph (bigraph) is used to obtain the second feature. In this graph, vertices are queries and ULRs. Each edge (q, u) in the graph shows that at least one user has clicked on URL u, as the search result of query q. The weight of the edge shows the number of users' clicks on URL u after query q. All other queries that lead to click on u are nominated for recommendation for query q. We use $G(Q + U, E)$ to refer to the bipartite graph where Q and U are the set of all queries and URLs, respectively which aggregately construct the set of vertices and E is the set of edges in the graph. We use $w[q, u]$ to denote the weight of the edge between query q and URL u.
From the bipartite graph, we construct a URL vector for each query in which a URL u exists in the vector of query q (with a weight of $w[q, u]$), if (q, u) exists in the graph G. Just as the previous works [9, 17], in order to compute the similarity between two queries q_x and q_y from the bipartite graph, the following steps should be taken:

- The set of URLs that appear in the URL vectors of both q_x and q_y are extracted. We use CU_{xy} to refer to this set.
- If $CU_{xy} = \emptyset$, the similarity between q_x and q_y is assumed zero.
- Otherwise, the similarity between q_x and q_y is obtained using the following equation:

$$Sim_{bigraph}(q_x, q_y) = \frac{\sum_{i=1}^{|CU_{xy}|}(w[q_x, CU_{xy}(i)] + w[q_y, CU_{xy}(i)])}{\sum_{i=1}^{|U|}(w[q_x, u_i] + w[q_y, u_i])} \quad (2)$$

where $CU_{xy}(i)$ is the i^{th} URL in set CU_{xy} and u_i is the i^{th} URL in set U. Moreover, $w[q_x, CU_{xy}(i)]$ denotes the weight of the edge between query q_x and URL $CU_{xy}(i)$ in the bigraph.
Fig. 2 illustrates a sample "Query-URL" bipartite graph. The number over each link shows the weight of that link. Suppose we want to obtain the similarity between q_5 and other queries in this graph. As shown in the figure, users have clicked on URLs u_1, u_2 and u_5 after submitting query q_5. Now, all other queries that result to a click on any of these URLs are identified, i.e., queries in the set $\{q_1, q_2, q_3, q_6\}$. The URL vector for each of these four queries has at least one URL in common with the URL vector of q_5.

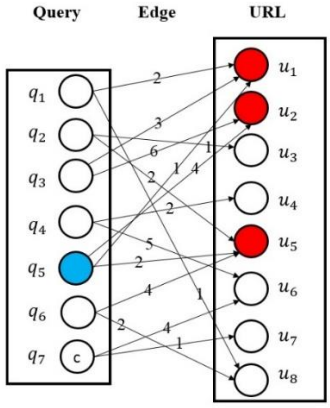

Fig. 2: A sample "Query-URL" bipartite graph

The similarity between q_5 and each query in the set $\{q_1, q_2, q_3, q_6\}$ is computed using Equation (2), while the rest of queries have zero similarity with q_5.
Since the average number of URLs clicked after each query is limited (e.g., about 3 in our dataset), computing the above similarity is not a difficult task. Although a click can be regarded as an indirect indication of a user's interest to the content of the corresponding URL, yet clicking on a URL can not necessarily denote the relevancy between queries. For example, a user may click on a URL by mistake or face with a click bait, while having no interest to that content. On the other hand, there are many relevant queries that have no common URL clicks and thus their similarity becomes zero, especially when we know that usually users click on a few top results. An additional feature is required to detect similar queries despite differences in their texts and clicks.

3-2-3- Top-k Search Results and their Ranks Feature

The third feature, for computing similarity between queries, is related to the top-k search results of query returned by the search engine. The idea is if two queries have several common search results, it is highly probable that they are relevant, even if they have no common terms or clicks. We only consider the results of the first search result page which typically contains 10 results (i.e., k=10). This is due to the fact that the first page usually receives much more traffic (click rate) than other pages: According to a recent research on Google search engine, the first page attracts 91.5%, the second page 4.8%, the third page 1.1%, and the forth page 0.4% of the traffic [29].

The ranking of search results is also an important factor. For example, by submitting the query "apple" in Google, a couple of first search results are related to the Apple Inc. and the next results are related to the apple fruit, while for the query "apples", the top results are related to the apple fruit. Although these two queries are lexically similar (after stemming), the intent of users for submitting them is quite different. This sample shows why the search results of queries and their ranks are beneficial to identify similar queries. In cases where users do not click on the same results or even do not click on any search result, this feature can be a proper measure to compute similarity between queries. We first weigh the search results of a query according to their ranks. The higher the rank of a URL is, the more its weight would be. The reason for this weighting is that the search engines rank the URLs based on their relevance to the submitted query; and a higher ranked URL receives more clicks from users [12, 30]. Therefore, the value of a URL decreases as its rank increases. We define the weight of a URL with rank i, with $w[i] = \frac{1}{2^i}$.

As before, we construct a result vector for each query in which a URL u exists in the vector of query q (with weight of $w[i]$), if u is the i^{th}-ranked search result for query q. To compute the similarity between two queries q_x and q_y, the following equation is used:

$$Sim_{top-10}(q_x, q_y) = \frac{1}{2}\sum_{i=1}^{m}\frac{w[r_x(i)] + w[r_y(i)]}{|r_x(i) - r_y(i)| + 1} \qquad (3)$$

where m is the number of common URLs in the top 10 search results of two queries q_x and q_y, $r_x(i)$ and $r_y(i)$ are the ranks of the i^{th} common URL with respect to query q_x and q_y, respectively. According to the above formula, as much as two queries have more common search results with similar ranks (especially in top ranks), they get a higher similarity score. For example, the presence of a same search result l in the first rank of both q_x and q_y implies a much higher score, compared with the case where the result l appears in different ranks for two queries. According to the weight value we assign to a search result in rank i, high-ranked common results can significantly boost the similarity score. As we explained before, the rationale is that the rank of each URL indicates the amount of its relevance to the submitted query.

3-2-4- Final Similarity

To get the final similarity between two given queries q_x and q_y, the values of the above three features are combined linearly as follows:

$$Similarity(q_x, q_y) = \alpha * Sim_{ngrams}(q_x, q_y) + \beta * Sim_{bigraph}(q_x, q_y) + \gamma * Sim_{top-10}(q_x, q_y) \qquad (4)$$

The sum of α, β and γ parameters is equal to one (i.e., $\alpha + \beta + \gamma = 1$). Therefore, the result of Equation (4) is always between 0 and 1, as the value of all three features is also between 0 and 1. The setting of these hyper-parameters is explained in Section 4. In order to reduce the number of incorrect recommendations, we use a minimum similarity threshold θ such that if the final similarity between two queries is smaller than θ, we assume they have a zero similarity, i.e., they are irrelevant. Based on our experiments, we use $\theta = 0.01$.

3-2-5- Medoid-based Clustering

In the last step of the offline phase, we use the final similarity feature to cluster our training queries with a *k*-medoids algorithm. In contrast to the *k*-means algorithm which calculates the means of points as centroids, *k*-medoids chooses points themselves as centroids. This is more compatible with our notion of pairwise similarity defined in Equation (4), since we do not assign any feature to each individual point (i.e., two queries cannot be averaged, they can only be compared with respect to the similarity features).

The most common realization of *k*-medoids is the Partitioning Around Medoids (PAM) algorithm [31] which works effectively for small datasets, but does not scale well for large datasets due to its time complexity. There have been some efforts in developing fast algorithms for *k*-medoids clustering, e.g., with sampling and randomization techniques. In this paper, we use a simple and fast algorithm for *k*-medoids clustering that works in three steps [32]:

1. Select initial medoids: Calculate the distance between every pair of training points, select *k* most middle points as initial medoids using a local heuristic method in [32].
2. Update medoids: Find a new medoid of each cluster, which is the point minimizing the total distance to other points in its cluster.
3. Assign objects to medoids: Assign each point to the nearest medoid, calculate the sum of distances from all points to their corresponding medoids. If the sum is equal to the previous one, then stop the algorithm. Otherwise, go back to Step 2.

The above algorithm can take a significantly reduced time in computation with comparable performance, against the PAM [32]. We use this algorithm to cluster all training queries, using the inverse of final similarity in Equation (4) as the distance between two queries:

$$Distance\ (q_x, q_y) = \frac{1}{Similarity\ (q_x, q_y)} \quad (5)$$

After clustering, we omit clusters containing only one query. As a result, at most 5% of queries are removed from our dataset, i.e., we have no recommendation for them. The remaining clusters may comprise a lot of queries which are not necessarily pairwise relevant, but it is very likely that they belong to the same category of information. For example, all queries about films, actors, and actresses may rest in a single cluster. Based on detected clusters, we construct three reverse lookup tables which will be used later to identify the corresponding cluster of an online (test) query:

- **N-gram lookup table:** For each 1/2/3-gram g and each cluster c, it shows how many times g is repeated in queries of cluster c.
- **Click lookup table:** For each URL u and each cluster c, it represents how many times u is clicked following queries in cluster c.
- **Result lookup table:** For each URL u and each cluster c, it indicates how many times u is returned as a search result of queries in cluster c.

The update of clusters is done periodically, e.g., in a per-day or per-week basis, through re-executing the above algorithm on the updated set of queries. During the update process, we can boost training queries based on the time of their occurrences, to put more emphasis on the recommendation of recent queries. We can also remove queries that have not been repeated for a while from the dataset.

It is worth mentioning that since we run the clustering algorithm in an offline environment, its running time does not affect the response time of test queries. In other words, we try to perform most time-consuming and complicated tasks in an offline manner to make our recommendations for online queries as soon as possible.

3-3- Online Randomized Query Recommendation

Our model is now trained and ready for query recommendation. In the online phase, when a test query q is submitted to the system, the closest cluster to q is found using the inverse lookup tables built in the clustering step. To do so, we first create an N-gram vector containing all 1/2/3-grams in q and also a URL vector having the first 10 search results of q. For each element in the N-gram vector, we search the N-gram lookup table and determine the frequency of occurring g in queries of cluster c as the score of cluster c. For the purpose of normalization, all cluster scores are divided by the maximum score. Similarly, for elements in the URL vector, we separately search the click and the result lookup tables to get the corresponding scores. The closest cluster to query q is the one with the highest total score with respect to the sum of above three scores. In rare cases where the score of top clusters are very close, we can select 2 or 3 clusters as the closest clusters to find relevant queries within them.

After finding the closest cluster which we call it designated cluster or dc in brief, we extract n most similar queries, in terms of the final similarity feature, from that cluster. In our work, we find top 10 similar queries (i.e., $n = 10$), and if the similarity between q and any of these queries is below the threshold θ, we ignore that query. To incur an acceptable query response time, we have to extract similar queries in a timely fashion. Since the number of queries in cluster dc can be large, calculating the similarity between q and every query in dc may be costly. We here devise a randomized nearest neighbor algorithm for approximating n most similar queries in regard to query q. This algorithm works iteratively as follows:

1. Initialization: Randomly select n queries from the designated cluster dc and put them in set S as the set of seed queries.

2. Distance calculation: Calculate the distance between q and every query in S, based on the distance metric in Equation (5).

3. Candidate selection: Sort all queries in S in ascending order, based on their distance from q, and put the first n queries in new set C (as the set of candidate queries).

4. Exploration: If new set C is different from the old C (obtained in the previous iteration), then for each query p in new C, greedily add n most similar queries with respect to p from cluster dc to set S and go to Step 2 for the next iteration. Otherwise, return new C.

The exploration step in the above algorithm is straightforward, as we do not perform any distance calculation or even sorting in this step. This is because the distance between every pair of queries in each cluster is calculated in the offline phase and we exactly know a priori which queries are the most n similar queries with respect to a query p in the designated cluster.

The advantage of the above algorithm is that it performs Step 2 and 3 for a limited number of queries in the designated cluster. In the worst case, it may go over all queries in dc or may return n queries that are not necessarily most similar to q. However, because of both randomized and greedy natures of the algorithm, it usually finds most similar queries after 2 or at most 3 iterations, even for very large clusters that contain thousands of queries. The results of our experiments on this algorithm is presented in the next section.

4- Evaluation

In this section, we explain the details of our experiments and present the evaluation results. All recommendation methods are implemented in Java with its native data structures such as ArrayList and LinkedList. To speed up the similarity computation and clustering process, we load all data directly in RAM without using any traditional databases. Our experiments are conducted on a server equipped with an Intel® Xeon® Processor X5650 and 32GB DDR3 RAM.

4-1- Experimental Dataset

To assess the efficiency of the recommendation algorithms, we use two separate datasets, from Parsijoo (the first Persian search engine) and AOL search engine logs[1]. The Parsijoo dataset consists of 7-days log of user activities from May 12 till May 18, 2018 and AOL search engine log contains web queries in a period of one month between March 1 and March 31, 2006. The statistics of two datasets are presented in Table 1. There is no record belonging to DDoS attacks in the Parsijoo log, as these attacks are abandoned by the front firewall. Moreover, we remove spam queries and click spams from our log as much as possible [33, 34].

Table 1: Statistics of Parsijoo and AOL datasets

Dataset	Time period	Number of queries	Number of queries led to a click	Number of unique queries
Parsijoo	7 days	932394	521542	246450
AOL	31 days	12138555	6484875	3382951

4-2- Evaluation Model

Our evaluation model is similar to the model used in previous works [17]. We first randomly partition our datasets into two roughly equal-sized parts and then, analyze the tuning of hyper-parameters, including α, β and γ in Equation (4), using the first part as the validation dataset. Then, we evaluate different algorithms using the other part as the evaluation dataset. We randomly select 1000 queries from the evaluation dataset as test queries and all remaining queries are used as training queries to train the model and cluster queries[2]. We perform the above steps for Parsijoo and AOL datasets separately. In order to see how robust our method is, we use different ratios of evaluation dataset (i.e., 90%, 70%, 50% and 30%) for training. For instance, when R=70%, we only inject 70% of the evaluation dataset for *k*-medoids clustering (the default ratio is R=90%). Also, the default number of recommendations is n=10.

As the evaluation metric, we use the precision which is defined as follows:

$$precision = \frac{TP}{TP+FP} \quad (6)$$

where TP (True Positives) denotes the number of relevant queries that are correctly recommended as relevant, and FP (False Positives) indicates the number of irrelevant queries that are wrongly recommended as relevant. In order to count TP and FP, we manually investigate the top results of recommendation algorithms for each test query to find out how many of them are relevant and irrelevant, respectively.

We also use the important p@10 metric as the number of correct recommendations made for each query over $n = 10$:

$$p@10 = \frac{TP}{10} \quad (7)$$

To understand the difference between two metrics, consider an example where a recommendation algorithm returns just one query and this query is relevant. The precision is 1, while the p@10 is 0.1. The results of precision and p@10 are averaged over all test queries. The running time and memory usage of the proposed method is calculated according to the executions on test queries.

4-3- Evaluation Results

4-3-1- Tuning of Parameters

To assess the impact of three hyper-parameters in Equation (4) and get the best configuration, we use an exhaustive grid search in the range [0,1] for α, β and γ parameters with step size of 0.1. We use the validation dataset for hyper-parameter tuning. Table 2 shows the results of precision and p@10 obtained in our method with different values of α, β and γ for the Parsijoo dataset. We separately tune our method for the AOL dataset as well. The highest precision and p@10 is achieved at $\alpha = 0.5$, $\beta = 0.3$ and $\gamma = 0.2$. For the AOL dataset, the best configuration is gotten at $\alpha = 0.4$, $\beta = 0.3$ and $\gamma = 0.3$. The N-gram feature reflects the pure textual similarity between queries and hence, it has a higher coefficient than other two features. On the other hand, the bipartite graph feature is more important than the top-10 search results feature, since it is supported by users' clicks. Thus, we can generally conclude that $\alpha > \beta \geq \gamma$. It is interesting that whenever we ignore any of these three features, we get a very bad result, especially with regard to the p@10 metric. For example, the p@10 degrades by up to 45% in cases where one of α, β and γ parameters is set to 0, while the precision worsens about 15%.

[1] The 3-months search log of AOL is publicly available over the Internet, e.g., in [8].

[2] To remove noises, we only consider test queries that have been repeated at least 5 times by different users.

Table 2: The precision and p@10 results of our method obtained with different values of α, β and γ parameters from Parsijoo dataset

α	β	γ	precision	p@10
0.5	0.3	0.2	66.8%	63.9%
0.4	0.3	0.3	65.4%	62.7%
0.5	0.5	0	61.3%	54.5%
0.5	0	0.5	62.1%	57.5%
0	0.5	0.5	57.9%	48%
1	0	0	60.2%	47.6%
0	1	0	54.3%	42.3%
0	0	1	55.4%	33.1%

Although the tuning of hyper-parameters is dependent on the dataset, but their optimum setting is not influenced by choosing different ratios of evaluation dataset (R). More specifically, the best value of α, β and γ is roughly the same for two cases where R=90% and R=30%.

Fig. 3 shows the impact of choosing n, i.e., the number of recommendations, on the precision (n varies from 1 to 10). We conduct this experiment on both Parsijoo and AOL datasets. The best result is achieved at $n = 1$. As the number of recommendations increases, the recommended queries become less similar to the user's submitted query, from the view point of three features, and thus, the number of false positives grows. The precision decreases about 30% when we change n from 1 to 10. For large values of n, the precision does not change much, as our algorithm can rarely add any more recommendation for a significant portion of queries (e.g., the precision at $n = 9$ is only 1% higher, compared with $n = 10$). The precision of recommendation for AOL is on average 8% higher than that of Parsijoo, especially for larger values of n. The main reason behind this phenomenon is that the amount of queries and users in the AOL dataset is an order of magnitude larger than queries in Parsijoo. Thus, the likelihood of finding relevant queries for a test query is higher.

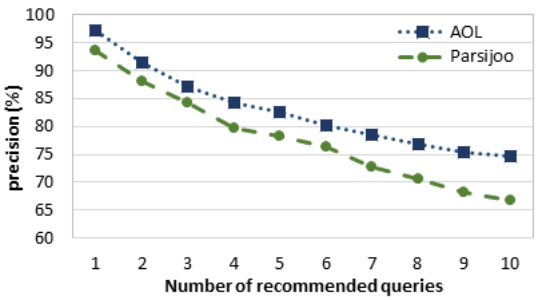

Fig. 3: The precision results of our method with respect to number of reommendations (n), obtained from AOL and Parsijoo datasets

Now, we analyze how the volume of training dataset affects the test results. The result of precision and p@10 for different ratios of Parsijoo dataset are shown in Fig. 4. As the amount of training data shrinks, the precision declines slightly. For example, the precision worsens only 11%, when R decreases from 90% to 30%. As much as we reduce the size of training dataset, the number of true relevant queries recommended for a given test query decreases. However, since we use a minium similarity threshold between relevant queries in Equation (4), the number of irrelevant queries recommended by our algorithm does not change much. But, the story is different when we pay attention to the p@10 metric. It is affected considerably by the value of R in a way that it drops by about one third when R falls to 30%. This is because our algorithm can barely find multiple relevant queries as we have not enough training queries. The trend of results in Fig. 4 remains roughly the same for AOL dataset. However, since the number of training queries is much larger in this dataset, the decrease in p@10 is not so devastating (the recall gets halved, as we reduce R from 90% to 30%).

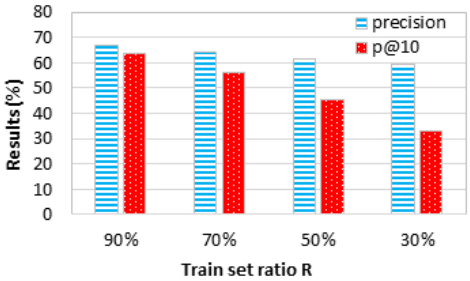

Fig. 4: The evaluation results of our method with respect to different evaluation dataset ratio *R*

The number of clusters in the *k*-medoids clustering algorithm impacts on both precision and response time of our method. Fig. 5 demonstrates the evaluation results of the proposed method obtained for Parsijo dataset when *k* raises from 50 to 500 with step size of 50. For smaller values of *k*, the precision is lower because it is more likely that irrelevant queries are placed in the same cluser. As a result, our randomized nearest neighbor algorithm probably returns some irrelevent queries in response to a submitted query. As much as we enlarge *k*, the precision improves as well. However, this improvement becomes negligible for very large values of *k*. On the other hand, the p@10 metric worsens when *k* goes beyond 350, because the number of relevant queries in clusters is reduced and our randomized algorithm fails in many cases to find 10 similar queries. Based on these results, we choose *k*=400 as the optimum value for our experiments.

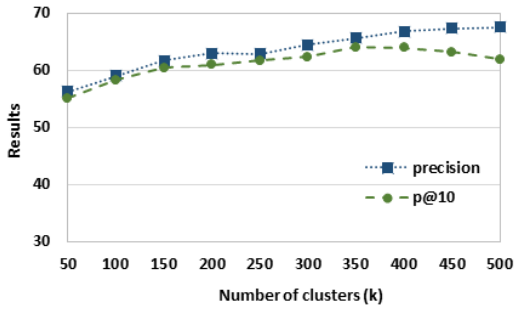

Fig. 5: The evaluation results of our method with respect to different number of clusters (*k*), obtained from AOL and Parsijoo datasets

The choice of *k* depends on the properties of dataset. Thus, we conduct the same experiment on the AOL dataset which leads to *k*=900 as the best choice. It is fascinating that the tunning of *k* is independent of the dataset ratio R (if the training queries are randomly selected from the whole dataset). From the perspective of response time, our randomized algorithm works faster when we increase the number of clusters, because it explores all candidate queries in fewer iterations. For example, the response time of the online algorithm gets halved when we change *k* from 50 to 500.

4-3-2- Evaluating the Online Algorithm

As discussed before, the randomized algorithm, proposed for approximating most similar queries in the online phase, makes a good tradeoff between the performance and response time of the recommendation system. We compare the performance results of this algorithm with the exhaustive algorithm which exactly finds the most similar queries through brute-force search in the designated cluster. The results of this comparison obtained from the Parsijoo dataset are presented in Table 3. The randomized algorithm produces the response 3 times faster than the exhaustive one at the cost of only 2% reduction in precision. As shown in the table, the average number of candidate queries investigated to find most similar queries is reduced by a factor of about 4. The results are improved further for bigger datasets like AOL, namely the response time decreases by a factor of 6 with only 3% loss in precision.

Table 3: Comparison between exhaustive and randomized recommendation algorithms

Algorithms	precision	p@10	#candidate queries	response time
Exhaustive nearest neighbor	68.1%	65.3%	672	615ms
Randomized nearest neighbor	66.8%	63.9%	164	201ms

4-3-3- Comparing Different Algorithms

Finally, we compare the performance of our method with three famous query recommendation algorithms introduced in [17], [13] and [24], which we refer to by "Bipartite graph", "Query clustering" and "KB-QREC", respectively. Table 4 shows the results of precision and p@10 obtained separately from Parsijoo and AOL datasets. Our proposed method, has a better precision in comparison with the other three methods, especially when we conside the Parsijoo dataset. More precisely, the proposed method outperforms KB-QREC by about 7%, and overcomes the bipartite graph and query clustering methods by more than 9% and 20%, respectively. From the viewpoint of p@10, the gap between the results of our method and other methods becomes even more apparent in a way that the improvement reaches to at least 23%. We achieve this advantage through utilizing various kinds of features simultaneously (from a linguistic feature to a user behavioral feature) to find similar queries. This property is more vital when we consider smaller datasets, since the only way to generate multiple recommendations is to catch similar queries from all different perspectives.

Table 5 compares the above four query recommendation methods with respect to the response time and memory usage parameters, obtained from Parsijoo and AOL datasets. To get the results, we submit 1000 queries to each recommendation method and then measure these two parameters for each method separately. First of all, since the size of AOL data is much larger than that of Parsijoo, we observe a considerably higher value for both parameters in all methods, when we take this dataset into account. The Bipartite graph is the simplest algorithm which is solely based the query-click graph. As a result, it yields the minimum value for both response time and memory usage. On the other hand, our method works with three different kinds of information to compute similarity features and hence, it has the worst memory usage. For instance, with Parsijoo dataset, it consume about 10%, 20% and 50% more memory than KB-QREC, Query clustering and Bipartite graph, respectively. This gab expands even further, when we consider the bigger AOL dataset.

Table 4: Comparison of different methods with respect to precision and p@10

Methods	Parsijoo		AOL	
	precision	p@10	precision	p@10
Query clustering [13]	51.5%	43.1%	62.2%	59.1%
Bipartite graph [17]	61.4%	38.8%	69.6%	62.6%
KB-QREC [24]	62.7%	48.8%	70.8%	67.2%
Our proposed method	66.8%	63.9%	74.7%	72.2%

In regard to the response time factor, the story becomes completely different. With Parsijoo dataset, the response time of our algorithm is about 10% and 16% shorter than KB-QREC and Query clustering methods, respectively (it still incurs a 10% longer response time than Bipartite graph). The response time of our method (in comparison with other methods) is improved more, when using AOL

dateset. More precisely, it outperforms KB-QREC and Query clustering by about 25% and 40%, respectively. This advantage is mainly due to the fast randomized algorithm we use in the online phase of our proposed method, as it tries to keep the set of candidate queries for finding most similar ones as small as possible.

Table 5: Comparison of different methods with respect to system parameters

Methods	Parsijoo		AOL	
	response time	memory usage	response time	memory usage
Query clustering [13]	386ms	51MB	1437ms	575MB
Bipartite graph [17]	288ms	44MB	789ms	409MB
KB-QREC [24]	354ms	56MB	1096ms	603MB
Our proposed method	321ms	62MB	814ms	681MB

5- Conclusions

In this paper, a novel method is proposed for query recommendation in search engines, using a combination of two query clustering and graph modeling approaches. As opposed to former methods, we take into account diverse features (query, click and result) to define similarity between queries and cluster them. In order to improve both precision and response time, we use an efficient *k*-medoids clustering algorithm as well as a new randomized nearest neighbor algorithm to find most similar queries. Evaluation results show that the proposed method outperforms some famous query recommendation methods with respect to precision and p@10 metrics. For example, the p@10 is improved by at least 23%, when compared with other counterparts.

References

[1] J. Wang, J. Z. Huang, J. Guo, and Y. Lan, "Query Ranking Model for Search Engine Query Recommendation", International Journal of Machine Learning and Cybernetics, Vol. 8, No. 3, 2015, pp. 1-20.

[2] M. Mitsui, "A Generative Framework to Query Recommendation and Evaluation", in ACM Conference on Human Information Interaction and Retrieval (CHIIR), 2017, pp. 407-409.

[3] X. Zhu, J. Guo, X. Cheng and Y. Lan, "More than Relevance: High Utility Query Recommendation by Mining Users' Search Behaviors", in 21st ACM International Conference on Information and Knowledge Management (CIKM), 2012, pp. 1814-1814.

[4] P. Melville and V. Sindhwani, "Recommender Systems", Encyclopedia of Machine Learning, Springer US, pp. 829-838, 2011.

[5] Y. Liu, J. Miao, M. Zhang, S. Ma and L. Ru, "How Do Users Describe Their Information Need: Query Recommendation based on Snippet Click Model", Expert Systems with Applications, Vol. 38, No. 11, 2011, pp. 13:847-13:856.

[6] W. Song, J. Z. Liang, X. L. Cao and S. C. Park, "An Effective Query Recommendation Approach using Semantic Strategies for Intelligent Information Retrieval", *Expert Systems with Applications*, Vol. 41, No. 2, 2014, pp. 366-372.

[7] Y. Song, D. Zhou and L. W. He, "Query Suggestion by Constructing Term-transition Graphs", in 5th ACM International Conference on Web Search and Data Mining (WSDM), 2012, pp. 353-362.

[8] Web Search Query Log Downloads. https://www.jeffhuang.com/search_query_logs.html (2018).

[9] Z. Zhang and O. Nasraoui, "Mining Search Engine Query Logs for Query Recommendation", in 15th International Conference on World Wide Web (WWW), 2006, pp. 1039–1040.

[10] J. R. Wen, J. Y. Nie and H. J. Zhang, "Clustering User Queries of a Search Engine", in 10th International Conference on World Wide Web (WWW), 2001, pp. 162-168.

[11] R. Baeza-Yates, C. Hurtado and M. Mendoza, "Query Recommendation using Query Logs in Search Engines", in International Conference on Extending Database Technology (EDBT), 2004, pp. 588-596.

[12] Y. Hong, J. Vaidya and H. Lu, "Search Engine Query Clustering using Top-k Search Results", in IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011, pp. 112-119.

[13] R. Chaudhary and N. Taneja, "A Novel Approach for Query Recommendation via Query Logs", International Journal of Scientific and Engineering Research (IJSER), Vol. 3, No. 8, 2012, pp. 1-6.

[14] L. Fitzpatrick and M. Dent, "Automatic Feedback using Past Queries: Social Searching?", in 20th Annual International Conference on Research and Development in Information Retrieval (SIGIR), 1997, pp. 306–313.

[15] K. Sugiyama, K. Hatano and M. Yoshikawa, "Adaptive Web Search based on User Profile Constructed without any Effort from Users", in 13th Conference on World Wide Web (WWW), 2004, pp. 675–684.

[16] B. J. Chelliah, R. Ojha, S. Semwal, P. Dobhal and C. Sahu, "Personalized Search Engine with Query Recommendation and Re-ranking", Journal of Network Communications and Emerging Technologies (JNCET), Vol. 8, No. 4, 2018, pp. 213-217.

[17] Q. Mei, D. Zhou and K. Church, "Query Suggestion using Hitting Time", in 17th ACM Conference on Information and Knowledge Management (CIKM), 2008, pp. 469–478.

[18] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis and S. Vigna, "The Query-flow Graph: Model and Applications", in 17th ACM Conference on Information and Knowledge Management (CIKM), 2008, pp. 609-618.

[19] A. Anagnostopoulos, L. Becchetti, C. Castillo and A. Gionis, "An Optimization Framework for Query Recommendation", in 3rd ACM International Conference on Web Search and Data Mining (WSDM), 2010, pp. 161-170.

[20] L. Bai, J. Guo and X. Cheng, "Query Recommendation by Modelling the Query-flow Graph", in Asia Information Retrieval Symposium (AIRS), 2011, pp. 137-146.

[21] F. Bonchi, R. Perego, F. Silvestri, H. Vahabi and R. Venturini, "Recommendations for the Long Tail by Term-Query Graph", in 20th International Conference on World Wide Web (WWW), 2011, pp. 15-16.

[22] J. Wang, J. Z. Huang, D. Wu, J. Guo and Y. Lan, "An Incremental Model on Search Engine Query Recommendation", Elsevier Neurocomputing, Vol. 218, 2016, pp. 423-431.
[23] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. G. Simonsen and J. Y. Nie, "A Hierarchical Recurrent Encoder-decoder for Generative Context-aware Query Suggestion", in 24th ACM International Conference on Information and Knowledge Management (CIKM), 2015, pp. 553-562.
[24] Z. Huang, B. Cautis, R. Cheng and Y. Zheng, "Kb-enabled Query Recommendation for Long-tail queries", in 25th ACM International Conference on Information and Knowledge Management(CIKM), 2016, pp. 2107-2112.
[25] R. Cheng, Y. Zheng and J. Yan, "Entity-based Query Recommendation for Long-tail Queries", ACM Transactions on Knowledge Discovery from Data, Vol. 1, No. 1, 2018, pp. 1-22.
[26] S. Kannan and V. Gurusamy, "Preprocessing techniques for text mining", International Journal of Computer Science & Communication Networks, Vol. 5, No. 1, 2014, pp. 1-7.
[27] Hunspell GitHub. https://github.com/hunspell (2018).
[28] H. Taghi-Zadeh, M. H. Sadreddini, M. H. Diyanati, and A. H. Rasekh, "A New Hybrid Stemming Method for Persian Language", Digit. Scholarsh. Humanit., Vol. 32, No. 1, 2017, pp. 209–221.
[29] Infront Webworks: Value of Organic First-page Results. https://www.infront.com/blogs/the-infront-blog/2015/6/17/value-of-first-page-google-results (2018).
[30] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski and G. Gay, "Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations in Web Search", ACM Transactions on Information Systems (TOIS), Vol. 25, No. 2, 2007, pp. 183-190.
[31] L. Kaufman, and P.J. Rousseeuw, "Clustering by Means of Medoids in Statistical Data Analysis based on the L1–norm and Related Methods", North-Holland, pp. 405–416, 1987.
[32] H. S. Park, and C. H. Jun, "A Simple and Fast Algorithm for K-medoids Clustering", Elsevier Expert Systems with Applications, Vol. 36, No. 2, 2009, pp. 3336-3341.
[33] T. Shakiba, S. Zarifzadeh, and V. Derhami, "Spam Query Detection using Stream Clustering", Springer World Wide Web (WWW), Vol. 21, No. 2, 2018, pp. 557–572.
[34] M. Fallah, S. Zarifzadeh, "Practical Detection of Click Spams using Efficient Classification-based Algorithms", International Journal of Information and Communication Technology Research, Vol. 10, No. 2, 2018, pp. 63-71.
[35] C. D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

Elham Esmaeeli-Gohari received the B.S. degree in Computer Engineering from Shahid Bahonar University, Kerman, Iran in 2013, and M.S. degree in Software Engineering from Yazd University, Iran, in 2017. Currently she is a Ph.D. Student in Isfahan University, Iran. Her research interests include Autonomous vehicles, Data fusion, Big data, Recommendation systems, Machine learning and Data mining.

Sajjad Zarifzadeh is an assistant professor of Computer Engineering at Yazd University, Iran. He received his Ph.D. in Computer Engineering from University of Tehran, Iran in 2012. His research is focused on Data analysis, Big data, Network security, Web and Internet services.

An Experimental Study on Performance of Text Representation Models for Sentiment Analysis

Sajjad Jahanbakhsh Gudakahriz
Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
Sa.jahanbakhsh84@gmail.com
Amir Masoud Eftekhari Moghadam*
Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
eftekhari@qiau.ac.ir
Fariborz Mahmoudi
Advanced Analytics Dept., General Motors Company, Warren, Mi, USA
fzmahmoudi@gmail.com

Received: 05/Feb/2020 Revised: 20/Jun/2020 Accepted: 15/Jul/2020

Abstract
Sentiment analysis in social networks has been an active research field since 2000 and it is highly useful in the decision-making process of various domains and applications. In sentiment analysis, the goal is to analyze the opinion texts posted in social networks and other web-based resources to extract the necessary information from them. The data collected from various social networks and web sites do not possess a structured format, and this unstructured format is the main challenge for facing such data. It is necessary to represent the texts in the form of a text representation model to be able to analyze the content to overcome this challenge. Afterward, the required analysis can be done. The research on text modeling started a few decades ago, and so far, various models have been proposed for performing this modeling process. The main purpose of this paper is to evaluate the efficiency and effectiveness of a number of commons and famous text representation models for sentiment analysis. This evaluation is carried out by using these models for sentiment classification by ensemble methods. An ensemble classifier is used for sentiment classification and after preprocessing, the texts is represented by selected models. The selected models for this study are TF-IDF, LSA, Word2Vec, and Doc2Vec and the used evaluation measures are Accuracy, Precision, Recall, and F-Measure. The results of the study show that in general, the Doc2Vec model provides better performance compared to other models in sentiment analysis and at best, accuracy is 0.72.

Keywords: Text Representation Models; Sentiment Analysis; Sentiment Classification; Ensemble Classifiers**.**

1- Introduction

With the advent of web 2.0 and the growing growth of social networks, the amount of free data produced by the users has reached an unpredictable amount [1]. With the high volumes of data, there is an increasing interest in the scientific community for creating systems that are capable of extracting information from the data. Sentiment analysis which is also known as opinion mining is a solution to the problem. Sentiment analysis refers to a series of methods, techniques, and tools for extracting opinions and emotional information from various types of texts for use in decision support and decision making systems [2-5].
Sentiment analysis employs a variety of techniques like natural language processing, machine learning, text analysis, statistics, and linguistics for analyzing and extracting information from texts [6]. In essence, a good sentiment analysis system might eliminate the need for polling, changes the methodology of traditional commercial research [7] and is very useful in realizing various goals such as identifying public attitude towards political movements, market intelligence, level of customer satisfaction, predicting the movie sales, and a lot more [8]. The wide range of internet-based technologies has increased the number of consumers who use the published opinions of others for making their purchasing decisions [9]. Therefore, various fields include product marketing, commercial intelligence, elections, military campaigns, political campaigns, medicine, software engineering, tourism, and cyberbullying use sentiment analysis [3, 4, 10-13].
Various steps are required for analyzing the input texts since these texts are obtained from different sources with different formats [14]. The process of sentiment analysis usually consists of three predefined steps which include data acquisition, preprocessing the text, and core process [15]. A general schematic of the steps is shown in Fig. 1.
The first step in sentiment analysis is data acquisition which involves obtaining and acquiring the text that can be used for extracting sentiments. The second step in sentiment analysis is the preprocessing of the texts. The messages and posts published on social networks are

* Corresponding Author

usually imperfect, noisy, poorly structured sentences, irregular expressions, typos, and non-dictionary expressions. Therefore, it is necessary to perform the preprocessing operation [16]. The third step in sentiment analysis is the core process, which is the extraction of sentiments about the selected entities from the text. In fact, at this step, the main task of sentiment analysis is carried out which involves various levels and dimensions.

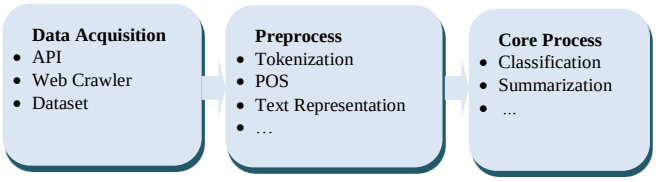

Fig. 1 General Steps of Sentiment Analysis

A fundamental problem of text mining tasks such as sentiment analysis is how to represent the text documents to make them mathematically computable. The raw data in the form of a series of symbols and words aren't proper for the machine learning algorithm; rather, it is required to transform the raw data into the form of a vector or any other format which can easily be processed [17]. Therefore, text representation is one of the important tasks in the preprocessing step of sentiment analysis and the efficiency of the tasks is significantly dependent on the selected model [18]. There are lots of models for text representation, that the most common of these are: TF-IDF, LSA, Word2Vec and Doc2Vec. Each of the models has different application and efficiency. The main purpose of the present study is to compare the efficiencies of these four common and well-known text representation models in sentiment analysis by existing ensemble classifier systems.

To this end, we employed an ensemble classifier system of sentiment classification to evaluate and compare the effectiveness of the four models. In this way, first, the texts are converted to the format of each of the reviewing models. Then it is given to the used ensemble classification system to classify these texts. Finally, the efficiency of the classification is evaluated.

The rest of this paper is organized as follows. Section 2 reviews in details the compared natural language representation models in this study. Section 3 presents the used ensemble classifier system in this study. The details of the experiments used for comparing the models and the results of these experiments are presented in section 4. Finally, Section 5 provides a summary and conclusions of the study.

2- Overview of Text Representation Models

In text mining and sentiment analysis, it is necessary to transform the selected text into a representation that is ready to be understood and analyzed by computer algorithms. The mapping from textual data to real-valued vectors is called feature extraction. The main challenge is finding the most valuable features by the models. The important matter is that before modeling the text, the main step is to break down the input string into words and then each word must be expanded and explained. The result will be a set of expressions corresponding to the input string. There are various models for doing this, in this section; we will discuss four models including TF-IDF[1], LSA [2], Word2Vec, and Doc2Vec. These models are commonly recognized models which are highly popular.

2-1- The TF-IDF Model

The TF-IDF model is one of the most common models of weighting words. In this model, each word is given a weight based on its frequency and the frequency of the word in other corpus texts [19]. In fact, the main goal of the weighting system is to show the importance of the word in the text. The weight of a word increases with its frequency in the text; however, it is controlled by the number of texts where the word appears. In other words, if a word appears in the majority of texts, it is most likely a common word, so it doesn't have a high value for the text evaluation. This method can be a good measure for determining the weight and importance of a word in a text. In fact, it shows how much a word can be unique and significant.

The value of TF-IDF is obtained by multiplying the term frequency in the document, $TF(t,d)$, by the inverse document frequency, $IDF(t,D)$, using Formula 1.

$$TFIDF(t,d,D) = TF(t,d).IDF(t,D) \quad (1)$$

The value of $TF(t,d)$ is the number of times that the term t frequencies in document d and is normalized using formula 2.

$$TF(t,d) = \frac{F(t,d)}{Max\ F(t_i,d)} \quad (2)$$

The $IDF(t,D)$ value is calculated using formula 3. In this formula, D indicates the set of documents, N is the size of D, and $|d \in D: t \in d|$ is the number of documents in the D that the term t appears.

$$IDF(t,D) = log\frac{N}{|d\in D:t\in d|} \quad (3)$$

Input Layer
1×V

[1] Term Frequency-Inverse Document Frequency
[2] Latent Semantic Analysis

2-2- The LSA Model

One of the methods for improving the text representation model is to use semantic information [18]. This idea is based on the assumption that words with lexical differences which are frequently used in similar documents have similar meanings. Semantic information can be computed using the LSA model. This model is a technique in text processing that can be used for modeling the conceptual relationship among a number of documents based on their set of words [20].

To represent a series of documents using the LSA model, a matrix (A) with the dimensions of $m \times n$ is created where n is the number of documents and m is the number of terms in all documents. Each column in A indicates a document (d) and each row indicates a term (t) (Fig. 2). There are various techniques for computing the values of the cells of this matrix, which indicate the frequency of the terms in the document. These methods include the TF-IDF method and the entropy method.

	Document 1	Document 2	…	Document n
Term 1				
Term 2				
…				
Term n				

Fig. 2 The Used Matrix in LSA Model

2-3- The Word2Vec Model

The Word2Vec is one of the models related to text processing where each term in the text is transformed into a vector [21]. This model uses a two-layer neural network and it's the most important idea in converting the words into vectors and transferring those to the vector space where the processing of words and documents can be easily done through machine learning tools. The output of the Word2Vec model is a vocabulary of words in the initial document along with their vector representation in an n-dimensional vector space. Related words or word groups are placed close to each other in this space, that this causes the semantic relationship between words to be maintained. The architecture of the Word2Vec model is shown in Fig. 3.

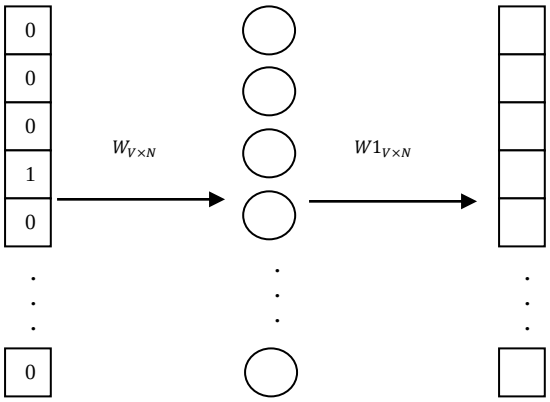

Fig. 3 The Architecture of Word2Vec Models [21]

2-4- The Doc2Vec Model

The Doc2Vec model is a developed model of the Word2Vec that applies to a sentence, paragraph, or text and creates a vector for it. The Word2Vec model produces high-quality word vectors; however, it doesn't have a specific method for combining these vectors to represent a text in the form of vectors. The Doc2Vec model was proposed for vector representation of a document. This model converts all the words in a sentence into a vector and puts the sentence label as a word inside this vector [22]. The vectors produced by the Doc2Vec model can be used to find semantic similarities between texts. The architecture of the Doc2Vec model is shown in Fig. 4.

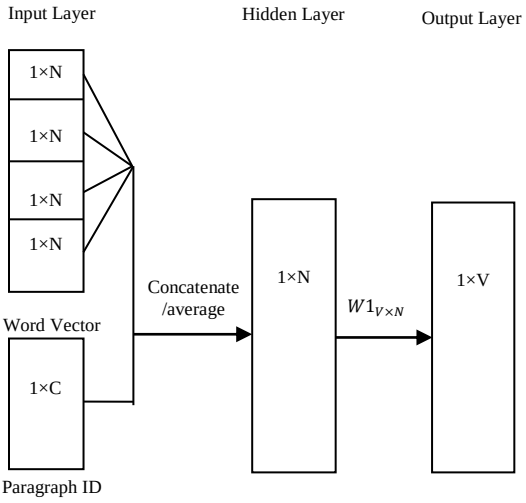

Fig. 4 The Architecture of Doc2Vec Model [22]

3- Ensemble Sentiment Classifier for Evaluation the Reviewed Models

Sentiment classification is the determining of the polarity of opinion texts in two or more classes. This can be done using machine learning methods, knowledge-based methods, or a combination of them. Machine learning methods provide a high level of accuracy, while knowledge-based methods provide a better generalization capability [15]. In machine learning methods, classification algorithms are used for separating sentiments and opinions into a number of separate classes.

In recent years, researchers suggested that ensemble classifiers provide a good capability for classification [23, 24, and 25]. Studies on using a combination of a number of basic classifiers show that using ensemble classifiers can improve the accuracy of sentiment classification [26].

In general, there are three common methods for combining classifiers including bagging, boosting, and random subspace [23]. In the bagging method, each one of the learners is created using independent random copies of the training dataset and the final results are calculated using simple majority polling. In the boosting method, basic learners are created from the weighted versions of the training dataset, which depend on the results of previous basic learners, and the final result is calculated using simple polling or weighted majority polling. In the random subspace method, basic learners are created based on the training dataset divided into a number of segments based on specific characteristics.

In this paper, we used the existing bagging method for sentiment classification. The used ensemble classifier system for evaluating text representation models is shown in Fig. 5. As can be seen from this figure, first the selected data for analysis is acquired from the dataset. Then, the necessary preprocessing tasks are performed on the dataset, and at the end of this step, the data is converted into a text representation model. Next, the prepared data is given to the classifiers, and the results obtained from the three classifiers are combined using majority polling. Finally, the result of the vote is evaluated.

Various classifiers were used in the experiments and finally based on the performance of these classifiers, Logistic Regression, K-NN and SVM classifiers were selected to be used in the ensemble system. Moreover, the preprocessing operations include tokenization, lemmatization, removing punctuation, ignoring usernames and hyperlinks, and replacing letters with a sequential repetition of more than twice in one word with two letters.

As shown in Fig. 5 and mentioned above, before sentiment classification, first the texts are converted to the format of the reviewed models, then the classification is done on the texts. Given that, the purpose of this paper is to evaluate the efficiency of four models, a fixed ensemble classifier is used, but the text display models are changed. In this way, the sentiment classification is done once with the TF-IDF model, once with the LSA model, once with the Word2Vec model and once with the Doc2Vec model. As a result, classification efficiency reflects the impact of each model on sentiment classification. In this way, the efficiency of each model in sentiment analysis is evaluated. In the next section, the experiments and the results are being explained.

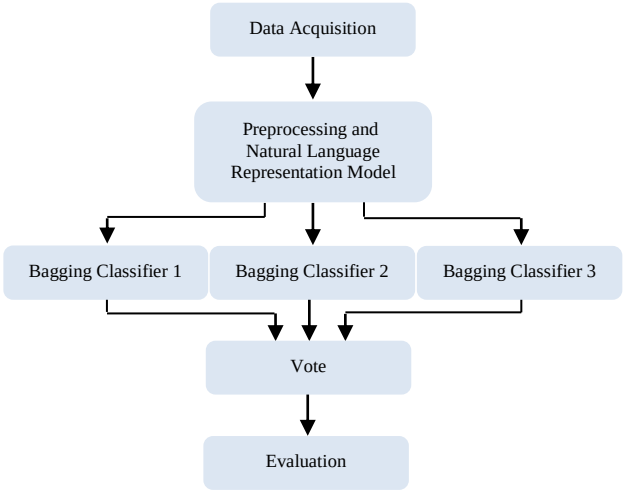

Fig. 5 The Used Ensemble Classifier

4- Experiments and Evaluation

In this section, the performed simulations for evaluating the selected models are presented. The evaluation includes measuring, comparing and analyzing the obtained results. To realize this goal, the ensemble classifier system explained in section 3 was applied on a number of datasets and the effects of various models on sentiment classification were investigated by applying these models in the preprocessing step. The simulations were done in Python 3 and the used datasets, metrics for measuring the efficiency, and the results of the simulations are presented below.

4-1- Datasets

To evaluate the performance of various models of text representation and comparing them to each other, it is required that the conditions of the experiments to be identical and the used data to be standard. Therefore, the experiments were performed on the existing four datasets known as HCR [27], Sanders [28], SemEval [29], and STS [30]. These datasets are common standard datasets for use in sentiment analysis and are used in many papers. They

are collected by API from tweets on Twitter. In order to improve confidence, these tweets were manually labeled based on the sentiments in each one. These datasets are related to different issues and they have two classes include positive and negative classes. From each of these datasets, 1000 tweets with positive and negative sentiments were selected, where 700 tweets were used for training and 300 tweets were used for testing. Among the selected tweets, half of them contained positive sentiments and the other half contained negative sentiments.

4-2- Performance Evaluation Measures

As mentioned above, the goal of the current study is to evaluate the efficiency of the selected text representation models by using them in sentiment classification. In fact, the model with the higher efficiency will increase the efficiency of the classifier system. Therefore, various measures must be selected for evaluating the used classifier system. Evaluating the efficiency of the sentiment classification system involves estimating how well a classifier system can predict the classes of texts with positive or negative sentiments. In the experiments, important measures were used for evaluating the efficiency of the sentiment classification system, and these include:

- Accuracy: This measure indicates what percentage of predictions is performed correctly and it is calculated using Formula 3.

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

- Precision: This measure indicates what percentage of positive results is actually positive, and it is calculated using Formula 4.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

- Recall: This measure indicates what percentage of the positives is predicted correctly, and it is calculated using Formula 5.

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

- F-Measure: This measure is the harmonic average of accuracy and recall, and it is calculated using Formula 6.

$$F - Measure = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (6)$$

As can be seen from Formula 3-6, to calculate the values of the evaluation measures, we have to extract the values for true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These values are calculated as follows:

- TP: The number of positive tweets correctly predicted as positive.
- TN: The number of negative tweets correctly predicted as negative.
- FP: The number of negative tweets incorrectly predicted as positive.
- FN: The number of positive tweets incorrectly predicted as negative.

4-3- Results of the Experiments

In this section, the results related to each one of the selected models are shown based on the measured metrics on the used datasets. Fig. 6 shows the results related to the accuracy of various datasets and models. As can be seen, the best performance among all the models and on all the datasets is achieved by the Doc2Vec model. On average, the accuracy value obtained for this model on all the datasets is 0.67. Moreover, the best value of accuracy obtained for different datasets is achieved by this model, and it is 0.72. On various datasets, other models have benefits over each other, and considering the average values for each model on all the datasets, after the Doc2Vec model, the LSA model with an average accuracy of 0.59 is achieved the best performance. Then, the Word2Doc model provided an average accuracy of 0.58. Finally, the weakest performance is achieved by the TF-IDF model, with an average accuracy of 0.57. Considering the average accuracy obtained from all the models on each dataset, on the HCR dataset, the average accuracy is 0.59; on the Sanders dataset, the average accuracy is 0.62; on the SemEval dataset, the average accuracy is 0.63; and on the STS dataset, the average accuracy is 0.56.

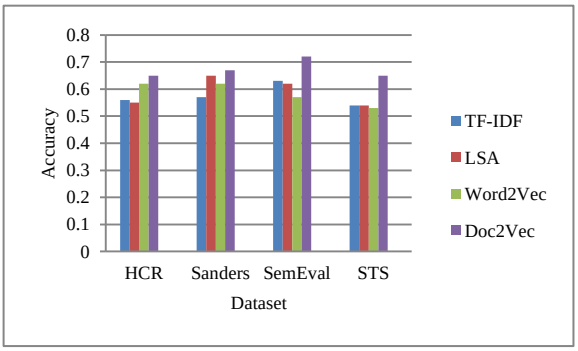

Fig. 6 Results for the Accuracy Measure

Fig. 7 shows the results for the precision measure on different datasets and models. The highest precision is achieved 0.72 by the LSA model on the Sanders dataset. On the other datasets, the Doc2Vec model has better performance, and the average precision for the Doc2Vec model on all the datasets is achieved 0.66, which is higher than the average values for all the other models. With regards to the average values of precision for each model on all the datasets, the Doc2Vec model is followed by the LSA model with an average precision of 0.60. Then, the TF-IDF model, with an average precision of 0.58 has the best performance, followed by the weakest performance by the Word2Vec model with an average precision value of 0.57. With regards to the average precision value for all the models on each dataset, on the HCR dataset, the average precision is achieved 0.58; on the Sanders dataset, the average precision is achieved 0.65; on the SemEval dataset, average precision is achieved 0.63; and on the STS dataset, average precision is achieved 0.56.

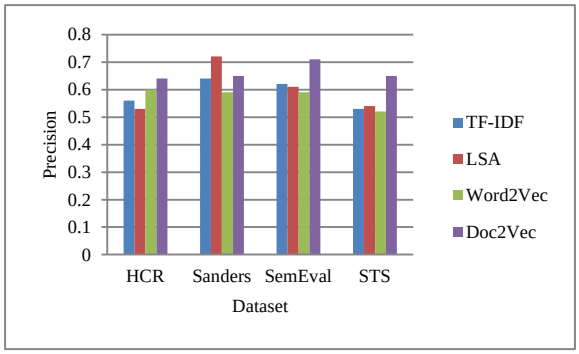

Fig. 7 Results for the Precision Measure

Fig. 8 shows the results for the F-Measure metric on different datasets and models. As can be seen, the Doc2Vec model achieves better performance compared to other models, and it has the best F-Measure value for various datasets equal to 0.72. On average, the F-Measure obtained for this model on all the datasets is 0.67. With regards to the average value for each model on different datasets, the Doc2Vec model is followed by the LSA model with an average F-Measure value of 0.58, followed by the Word2Vec model with an average F-Measure value of 0.57, followed by the TF-IDF model with the weakest performance and an average F-Measure value of 0.54. With regards to the average F-Measure value obtained for all the models on each dataset, on the HCR dataset the average F-Measure value is achieved 0.58; on the Sanders dataset the average F-Measure value is achieved 0.57; on the SemEval dataset, the average F-Measure value is achieved 0.64; and on the STS dataset, the average F-Measure value is achieved 0.56.

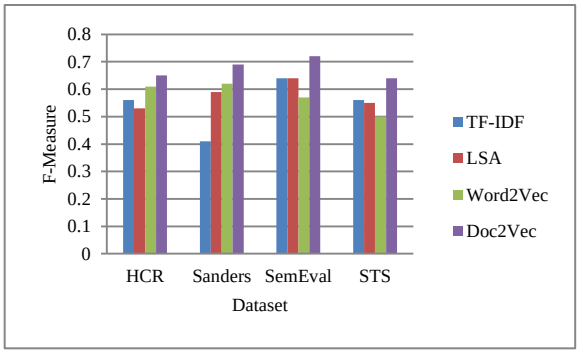

Fig.8 Results for the F-Measure Metric

Fig. 9 shows the results of the recall measure for various datasets and models. As can be seen, the best recall value for different datasets is achieved 0.73 for the Doc2Vec model on the Sanders dataset. The Doc2Vec model obtains the best performance among all the models and datasets, and on average, the recall value obtained for this model on all the datasets is achieved 0.68. With regards to the average value of recall for each model on various datasets, after the Doc2Vec model, the Word2Vec model with an average recall of 0.57 has the best performance. This model is followed by the LSA model with an average recall value of 0.56, and the TF-IDF model provides the weakest performance with an average recall value of 0.54. With regards to the average recall value obtained for all the models on each dataset, on the HCR dataset, the average recall is achieved 0.59; on the Sanders dataset, the average recall is achieved 0.54; on the SemEval dataset, the average recall is achieved 0.65; and on the STS dataset, the average recall is 0.56.

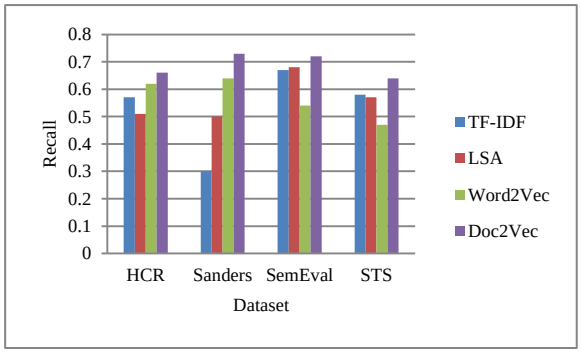

Fig. 9 Results for the Recall Measure

Based on Figs 6-9 and the results explained above, it can be concluded that the Doc2Vec model provides the best performance in sentiment analysis among the four tested models. Based on the calculated averages as well as the four measured metrics and considering the high

significance of accuracy and precision measures, it can be said that the LSA model has second performance in sentiment analysis. After this model, the Word2Vec model comes next in terms of performance and the weakest performance is achieved by the TF-IDF model. Moreover, based on each dataset and the overall average values obtained in the experiments, it can be said that the performance of all the models is better on the SemEval dataset compared to all the other datasets.

5- Conclusions

Text representation models play an essential role in sentiment analysis. Therefore, various models have so far been proposed for performing this task. These models have different efficiency in different applications, as well as many of these models are used in sentiment analysis researches. Therefor the main purpose of the current study was to compare the efficiency of common and famous text representation models in this field in sentiment analysis by experiments. For this goal, an ensemble classifier system was used for sentiment classification and at the preprocessing step of this system, each of the models including TF-IDF, LSA, Word2Vec, and Doc2Vec were used. Four different standard datasets of tweets were selected, and the efficiency of these models was evaluated on these datasets. The simulation results show that in general, the Doc2Vec model provides better performance compared to other models based on basic evaluation measures for sentiment classification. It seems that using this model in other applications of sentiment analysis can lead to better results since the classification of opinions is a primary task in sentiment analysis and the efficiency of classification has a significant direct impact on other possible operations as well.

References

[1] A. Montoyo, P. Martínez-Barco, and A. Balahur, "Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments", Decision Support Systems, Vol. 53, 2012, pp. 675–679.

[2] J. A. Balazs, and J. D. Velasquez, "Opinion Mining and Information Fusion: A survey", Information Fusion, Vol. 27, 2016, pp. 95–110.

[3] F. A. Pozzi, E. Fersini, E. Messina, and B. Liu, Sentiment Analysis in Social Networks (Chapter 1), Science Direct, 2017.

[4] V. M. Mika, G. Daniel, and K. Miikka, "The evolution of sentiment analysis—A review of research topics, venues, and top cited papers", Computer Science Review, Vol. 27, 2018, pp. 16-32.

[5] E. Cambria, "Affective computing and sentiment analysis", IEEE Intelligent Systems,Vol. 31, No. 2, 2016, pp. 102-107.

[6] P. Gupta, R. Tiwari, and N. Robert, "Sentiment Analysis and Text Summarization of Online Reviews: A Survey", in International Conference on Communication and Signal Processing, 2016, pp. 241-245.

[7] J. A. Balazs, and J. D. Velasquez, "Opinion Mining and Information Fusion: A survey", Information Fusion, Vol. 27, pp. 95–110, 2016.

[8] A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydar, "Detection of review spam: A survey", Expert Systems with Applications, Vol. 42, 2015, pp. 3634–3642.

[9] A. Qazi, R. Raj, G. Hardaker, and C. Standing, "A systematic literature review on opinion types and sentiment analysis techniques: tasks and challenges", Internet Research, Vol. 27, 2017.

[10] F. Bisio, L. Oneto, and E. Cambria, Sentiment Analysis in Social Networks (Chapter 5), Science Direct, 2017.

[11] S. Sun, C. Luo, and J. Chen, "A Review of Natural Language Processing Techniques for Opinion Mining Systems", Information Fusion, 2016.

[12] P. Sobkowicz, M. Kaschesky, and G. Bouchard, "Opinion mining in social media: Modeling, simulating, and forecasting political opinions in the web", Government Information Quarterly, Vol. 31, 2012, pp. 470–479.

[13] M. Thelwall, "Sentiment Analysis for Tourism", Big Data and Innovation in Tourism, Travel, and Hospitality, 2019, pp. 87-104.

[14] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, "A survey of sentiment analysis in social media", Knowledge and Information Systems, 2018.

[15] K. Ravi, and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications", Knowledge-Based Systems, Vol. 89, 2015, pp. 14–46.

[16] Z. Jianqiang, and G. Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis", IEEE Access, Vol. 5, 2017, pp. 2870 – 2879.

[17] L. W. Manod, P. J. Wiliam, and T. R. Michael, "Variable segmentation and ensemble classifiers for predicting dairy cow behavior", Biosystems engineering, Vol. 17, 2019, pp. 156-167.

[18] T. Brychcin, and M. Konopik, "Semantic spaces for improving language modeling", Computer Speech and Language, Vol. 28, 2014, pp. 192–209.

[19] A. Aizawa, "An information-theoretic perspective of tf–idf measures", Information Processing and Management, Vol. 39, 2003, pp. 45-65.

[20] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis", Journal of the American Society for Information Science, Vol. 41, 1990, pp. 391–407.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space", Proc. Workshop at ICLR, 2013.

[22] Q. Le, and T. Mikolov, "Distributed representations of sentences and documents", in Proceedings of the 31st International Conference on Machine Learning (ICML 2014), 2014, pp. 1188–1196.

[23] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: The contribution of ensemble learning", Decision Support Systems, 2013.

[24] I. Furxhi, F. Murphy, M. Mullins, and C. A. Poland, "Machine Learning Prediction of Nanoparticle In Vitro Toxicity: A Comparative Study of Classifiers and Ensemble-

Classifiers using the Copeland Index", Toxicology Letters,Vol. 312, 2019, pp. 157-166.
[25] I. Lopez-Gazpio, M. Maritxalar, M. Lapata, and E. Agirre, "Word n-gram attention models for sentence similarity and inference", Expert Systems With Applications, Vol. 132, 2019, pp. 1-11.
[26] N. F. F. Da Silva, E. R. Hruschka, and J. E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles", Decision Support Systems, 2014.
[27] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter polarity classification with label propagation over lexical links and the follower graph", in Proceedings of the EMNLP First workshop on Unsupervised Learning in NLP, 2011.
[28] W. Deitrick, and W. Hu, "Mutually enhancing community detection and sentiment analysis on twitter networks", Journal of Data Analysis and Information Processing, Vol. 1, 2013, pp. 19-29.
[29] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter and T. Wilson, "Semeval-2013 task 2: Sentiment analysis in twitter", in Proceedings of the 7th International Workshop on Semantic Evaluation, Association for Computational Linguistics, 2013.
[30] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision", CS224N Project Report, Stanford, 2009.

Sajjad Jahanbakhsh Gudakahriz received the B.S. degree in Computer Engineering from Azad University, Ardabil Branch, Iran in 2008, and M.S. degree in Computer Engineering from Azad University, Qazvin Branch, Iran, in 2011. Currently he is Ph.D. Candidate in Azad University, Qazvin Branch, Iran. His research interests include Sentiment Analysis, Sentiment Summarization, Data mining and Computer Networks.

Amir Masoud Eftekhari Moghadam was born in Tehran, Iran in 1964. He received B.E. and M.Sc. degree in computer engineering from Iran University of Science and Technology in 1992 and 1995, respectively and Ph.D. degree in computer engineering in 2002 from Science and Research branch, Islamic Azad University, Tehran, Iran. He is associate professor at the Islamic Azad University, Qazvin, Iran. His research interests include image processing, image/video coding, machine learning, data mining, Big data analysis and semantic video analysis.

Fariborz Mahmoudi received BS, MS, and Ph.D. degree in computer engineering. From 2002 to 2013, he had been an assistant professor in Electrical, Computer, and IT engineering faculty at the Qazvin Azad University, Iran. From 2013 until 2016, he had continued his researches in the image analysis lab of Henry Ford Health System (Detroit, MI, USA). Since Jan. 2017, he is a data scientist in General Motors company (Warren, MI, USA). His research interests include machine learning, machine vision, and information (image and text) retrieval. Dr. Mahmoudi has published more than 100 papers in the international scientific journals and conferences proceedings.

Using Residual Design for Key Management in Hierarchical Wireless Sensor Networks

Vahid Modiri
Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
va_modiri@srbiau.ac.ir
Hamid Haj Seyyed Javadi*
Department of Mathematics and Computer Science, Shahed University, Tehran, Iran
h.s.javadi@shahed.ac.ir
Amir Masoud Rahmani
Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
rahmani@srbiau.ac.ir
Mohaddese Anzani
Department of Mathematics and Computer Science, Shahed University, Tehran, Iran
anzani@shahed.ac.ir

Received: 25/Nov/2019 Revised: 16/Dec/2019 Accepted: 08/Feb/2020

Abstract

Combinatorial designs are powerful structures for key management in wireless sensor networks to address good connectivity and also security against external attacks in large scale networks. Many researchers have used key pre-distribution schemes using combinatorial structures in which key-rings, are pre-distributed to each sensor node before deployment in a real environment. Regarding the restricted resources, key distribution is a great engagement and challenging issue in providing sufficient security in wireless sensor networks. To provide secure communication, a unique key should be found from their stored key-rings. Most of the key pre-distribution protocols based on public-key mechanisms could not support highly scalable networks due to their key storage overhead and communication cost that linearly increasing. In this paper, we introduce a new key distribution approach for hierarchical clustered wireless sensor networks. Each cluster has a construction that contains new points or that reinforces and builds upon similar ideas of their head clusters. Based on Residual Design as a powerful algebraic combinatorial architecture and hierarchical network model, our approach guarantees good connectivity between sensor nodes and also cluster heads. Compared with similar existing schemes, our approach can provide sufficient security no matter if the cluster head or normal sensor node is compromised.

Keywords: Wireless sensor networks; Key pre-distribution; Residual Design; Hierarchical network model.

1- Introduction

A wireless sensor network (WSN) is a group of a majority of sensor nodes with limited resources such as storage capacity, energy, computational resources, and radio transmission range. We focus on the mechanisms of organizing secret keys between sensor nodes known as the key management problems. Some researches focused on the key management problem [1-6]. The key pre-distribution scheme (KPS) is a practical way to handle key management problems in wireless sensor networks where a set of specified keys is-preloaded in the nodes before deployment. There are three types of random, deterministic, and hybrid for key pre-distribution schemes. In random schemes, key-rings are randomly chosen from the main key-pool. In
deterministic key pre-distribution schemes, key-rings are pre-loaded to sensor nodes based on a particular arrangement. In hybrid schemes both the mentioned methods are used. Combinatorial designs are one of the most important approaches used to consider a deterministic KPS. Based on combinatorial design, some approaches were introduced, e.g., [7-13]. All of these proposed schemes considered in a flat network. In such networks, all of the available sensor nodes have the same specifications such as battery life, memory capacity, and computational power. However, flat networks are suitable in terms of efficiency and simplicity for sensor applications. Effective hierarchical key management is required to improve these parameters. Some researchers provided "hierarchical architecture" as their chosen architecture for wireless sensor networks [14-17].

* Corresponding Author

Table 1: Residual Design for (7,3,1)-SBIBD

$c_1 = \{4,5,6,7\}$	$c_2 = \{2,3,6,7\}$	$c_3 = \{2,3,4,5\}$	$c_4 = \{1,3,5,7\}$	$c_5 = \{1,3,4,6\}$	$c_6 = \{1,2,5,6\}$	$c_7 = \{1,2,4,7\}$
{4,5}	{2,3}	{2,3}	{1,3}	{1,3}	{1,2}	{1,2}
{6,7}	{6,7}	{4,5}	{1,5}	{1,4}	{1,5}	{1,4}
{4,6}	{2,6}	{2,4}	{1,7}	{1,6}	{1,6}	{1,7}
{5,7}	{2,7}	{2,5}	{5,7}	{4,6}	{2,6}	{2,4}
{4,7}	{3,7}	{3,4}	{3,7}	{3,4}	{2,5}	{2,7}
{5,6}	{3,6}	{3,5}	{3,5}	{3,6}	{5,6}	{4,7}

Recent researches show that in the comparison between the two main architectures, hierarchical architecture has better scalability and performance in large-scale wireless sensor network.

In a hierarchical WSN, each node is assigned a role of either Cluster Head (CH) or Cluster Node (CN), considering its capabilities. In each cluster, the node with more capability and resource is considered as CH. The CNs of each cluster collect data, communicate with their neighboring nodes directly or via a multi-hop communication path, and finally, send collected data to their resource-rich CH. In this paper, we present a secure architecture for sensor nodes by proposing a new hierarchical key management method. In our approach, we take advantage of combinatorial designs, and in particular Residual Design [11], for deterministic key pre-distribution. The structure of Residual Design allows each CN to share a common key with its CH, which makes direct communication of each CH with its CNs.

We conduct both analytical and experimental evaluations, considering different criteria: network scalability, connectivity, and resilience against nodes attacks. Network scalability is the maximum number of cooperating nodes supported in a WSN. Network connectivity is the probability of sharing at least a unique key between any two neighboring nodes. Network resiliency is the ability to resist of the network against the compromising of sensor nodes. We show that our approach improves network security and scalability, while provides a reasonable key share probability between the sensor nodes. The structure of this paper is organized as follows. We discuss the background and related works in section 2. The model and assumptions are presented in section 3. Our new approach named RDH is proposed in section 4. Analytical analysis and the simulation of different schemes are compared in part 5. The experimental results are provided in section 6. In the last Section, conclusion and future work directions are outlined.

2- Background and related works

2-1- Background

Definition 1. In a set system $(X, \mathcal{A})$, the set X consists of v elemets and $\mathcal{A}$ is subsets of X known as blocks. The combinatorial design theory is the technique of arranging members of $\mathcal{A}$ into words, arrays, or subsets based on predefined rules. Balanced Incomplete Block Design (BIBD) is one such design. In (v,k,λ) -BIBD or (v,b,r,k,λ)-BIBD, there are v elements repeated r times in total b blocks of length k. Every two points exist in exactly λ of b blocks. In BIBD, we have: $bk = vr$ and $\lambda(v-1) = r(k-1)$.

A BIBD is symmetric (v,k,λ)-SBIBD when $b = v$ and consequently $r = k$ [18].

Definition 2. Let (V,B) be a symmetric BIBD with $V = \{x_1, \dots, x_v\}$ and $B = \{B_1, B_2, \dots, B_v\}$. For any i, the blocks $B_1 \backslash B_i, B_2 \backslash B_i, \dots, B_i \backslash B_{i-1}, B_{i+1} \backslash B_i, \dots, B_v \backslash B_i$ are constructing a $(v-k, v-1, k-\lambda, \lambda)$-BIBD of the point set $X \backslash B_i$. This generalization is called Residual Design [11]. To construct Residual Design or RD structure with the symmetric BIBD $(q^2+q+1, q+1, 1)$, the point set of each class forms a BIBD with parameters$(v,b,r,k,\lambda) = (q^2, q^2+q, q+1, q, 1)$.

Example 3. Consider (7,3,1)-SBIBD with the following blocks: $B_1 = \{1,2,3\}$, $B_2 = \{1,4,5\}$, $B_3 = \{1,6,7\}$, $B_4 = \{2,4,6\}$, $B_5 = \{2,5,7\}$, $B_6 = \{3,4,7\}$, $B_7 = \{3,5,6\}$. Then over the point set $C_1 = \{4,5,6,7\}$, the RD sets $B_2 \backslash B_1 = \{4,5\}$, $B_3 \backslash B_1 = \{6,7\}$, $B_4 \backslash B_1 = \{4,6\}$, $B_5 \backslash B_1 = \{5,7\}$, $B_6 \backslash B_1 = \{4,7\}$, $B_7 \backslash B_1 = \{5,6\}$ are the blocks of a $(4,6,3,2,1) - \text{BIBD}$. The remaining blocks can be observed in table 1.

In Residual Design, each class C_i consists of the elements $X \backslash B_i$.

2-2- Related works

For the sake of blindness in the topology of the hierarchical network before the deployment, we have to store keys into memories of sensor nodes via a pre-distribution scheme

before deployment. Symmetric approaches are good for wireless sensor networks due to their low resource requirements such as resource and energy consumptions [19].

In [6], Eschenauer and Gligor proposed the RKP approach as the basic random key pre-distribution scheme. In this scheme, a random key-ring of size k is pre-loaded in each sensor node. In [7], Chan et al. proposed an enhancement of RKP for the network resiliency called q-composite. In this algorithm, two adjacent sensor nodes can establish a connection if there would be at least q common keys. In [8], Camptepe and Yener introduced symmetric key pre-distribution designs using parameters $(q^2+q+1, q+1, 1)$-SBIBD. Their basic approach had full connectivity but the resilience was not good enough.

Then they provided a hybrid design. In the hybrid design, the complement of each block is generated and the key-rings of extra nodes were provided from those blocks. In the hybrid design, the resilience and also the scalability was improved. Lee and Stinson [9] use the transversal design (TD) for key pre-distribution. In the q-composite scheme, if two nods have $q \geq q'$ keys in common, they can communicate. In [14], Javanbakht et al. proposed a key pre-distribution scheme for a clustered heterogeneous WSN using transversal designs denoted by TDH. Zhang et al.[10] proposed a secure efficient hierarchical key management scheme (SEHKM) for wireless sensor networks. They introduced an assistant node near the cluster head to improve the resiliency and reduce the usage of the resources. In [11], Modiri and Anzani proposed a robust highly scalable key pre-distribution approach based on a new combinatorial algebraic named Residual Design or RD. In RD, good results in terms of connectivity and also very high network scalability are achieved. Also, they introduced a new modification of RD called RD* to improve the resilience of their first scheme.

Anzani and Modiri in [12] proposed the merging hybrid symmetric design to solve the connectivity problem of hybrid symmetric design presented in [8]. They achieved better connectivity results compared with similar related schemes. In MGHS, d blocks of SBIBD were merged instead of constructing a complimentary design in the main scheme.

In [15], Cheng et al. proposed an improved key distribution mechanism for hierarchical and large scale wireless network models. This approach with low communication overhead denoted as IKDM used a bivariate polynomial key-generation mechanism to guarantee that each two clusters can communicate via a unique pairwise key. In [16], Zhang et al. proposed a model in hierarchical networks based on the probability of nodes capture. Depend on nodes attack probabilities, the arrangement of hierarchical nodes could be changed, so that the nodes with smaller probability, tends to be located as cluster heads. The key management scheme uses the mechanism of Exclusion Basis Systems to update the key-rings among clusters.

Albakri et al.[17] introduced a deterministic hierarchical clustered key distribution scheme based on polynomials. They find a pairwise key between any nodes in a cluster and also between the sink node and each cluster heads. In this scheme, to reduce the risk of security attacks, a novel probabilistic feature is used.

3- Model and assumptions

3-1- Network model

Wireless sensor networks are classified as being either flat or hierarchical configuration. In flat architecture, almost entire nodes play the same role, whereas, in a hierarchical sensor network, their roles are different.

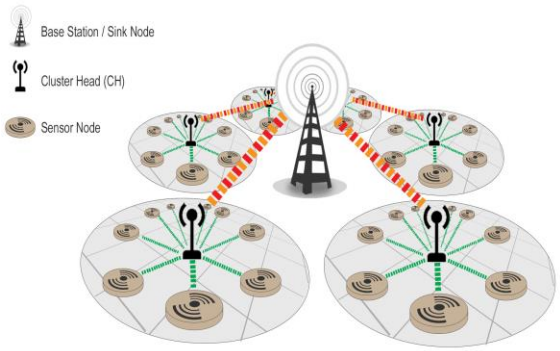

Figure 1: Hierarchical wireless sensor network architecture in our network model. We have q^2+q+1 clusters each consists of a cluster head and q^2+q sensor nodes.

In several aspects such as energy constraints and high network scalability, the hierarchical network model has more operational benefits. In many applications of wireless sensor networks, connectivity between all sensors is not prerequisite and wireless sensors just transmit data to the nodes with similar capabilities [20, 21]. In this work, we concentrate on highly scalable wireless sensor networks in hierarchical network architecture. In figure 1., our network

model is illustrated. The architecture has three types of wireless sensor nodes: Sink Node or Base Station (BS), Cluster Head (CH), and Cluster Node (CN).

Sink node: Sink node or base station is a powerful device at the top or in the center of the network. It has unlimited resources such as memory storage, communication power, computation, and a high range of radio transmission to access entire nodes in the network.

Cluster head: Cluster heads have more powerful resources and wider radio transmission range than a normal sensor node. They can transmit data between their members and the sink node.

Sensor node: Sensor nodes are cheaper and have lower resources. They have limited battery power, data processing capability, memory size, and also not wide radio transmission range.

In our model, any sensor node can communicate with its cluster head directly; also may communicate with its sibling in the related cluster headset. In the deployment, the sensor network is partitioned into several clusters known by their cluster heads. The clustering process is performed by some existing clustering algorithms [22, 23]. In this model, each cluster is composed of a cluster head and a set of sensor nodes. As shown in Fig. 1, the sensor nodes directly send data to the related cluster head and also would communicate with other sensor nodes inside the current partition.

Algorithm 1: RDH

Require: N {Total number of nodes}

1. Find the minimum prime power q that satisfies$(q^2 + q + 1)^2 \geq N$.
2. Generate the base symmetric design with parameters$(q^2 + q + 1, q + 1,1)$.
   - v objects $P = \{a_1,a_2,...,a_v\}$.
   - b blocks $B = \{B_1,B_2,...,B_b\}$ of size $q + 1$.
3. Generate $c' = (q^2 + q + 1)$ blocks for classes c_i and $b' = (q^2 + q + 1)(q^2 + q)$ blocks for constructing internal members of Residual Design from the base symmetric design:
   - Blocks $C_i = P\backslash B_i$ where $i = 1, \ldots, q^2 + q + 1$.
   - Blocks $B_{ij} = B_i\backslash B_j$ where $i, j = 1, \ldots, q^2 + q + 1$.
4. Assign C_i blocks to cluster heads and B_{ij} to sensor nodes.

4- Our proposed approach

Considering the hierarchical network model and the Residual Design(RD), we propose a secure scalable key distribution approach for clustered hierarchical wireless sensor networks based on Residual Design. We denote our new approach as RDH. Clusters of the WSN are built based on the RDH algorithm in the offline area. The clustering procedure in our key pre-distribution approach is based on the distribution of the required key-rings on cluster heads and cluster nodes before deployment.

This process is performed in 2 phases: the key pre-distribution phase and shared-key discovery phase. In the key pre-distribution phase, we decide on choosing N secure key-rings from a key pool and pre-load them into cluster heads and also all sensor nodes before they are deployed. Given a wireless network size N, we consider the minimum power q in which $(q^2 + q + 1)^2 \geq N$ to construct a $(q^2 + q + 1, q + 1,1)$-SBIBD. Based on this SBIBD, we construct the Residual Design with $c' = (q^2 + q + 1)$ classes in which the block set $C_i = \{P\backslash B_i\}$ of size q^2 is assigned to each class. Each class has $(q^2 + q)$ blocks $B_{ij} = \{B_i\backslash B_j\}$ inside, therefore all classes could generate totally $b' = (q^2 + q + 1)(q^2 + q)$ blocks of size q. Then the blocks C_i are assigned to c' cluster heads and also the blocks B_{ij} are assigned to $b' = N - c'$ Cluster nodes.

The construction algorithm of our proposed approach for hierarchical wireless sensor networks is described in the Algorithm 1.

After the key pre-distribution phase, the nodes are randomly deployed in the environment and the $(q^2 + q + 1)$ clusters are built. Clusters are built based on different criteria such as location, mission type, and communication range [24].

In the shared key-discovery phase, any two cluster heads or

any sensor node and its cluster head must find at least a

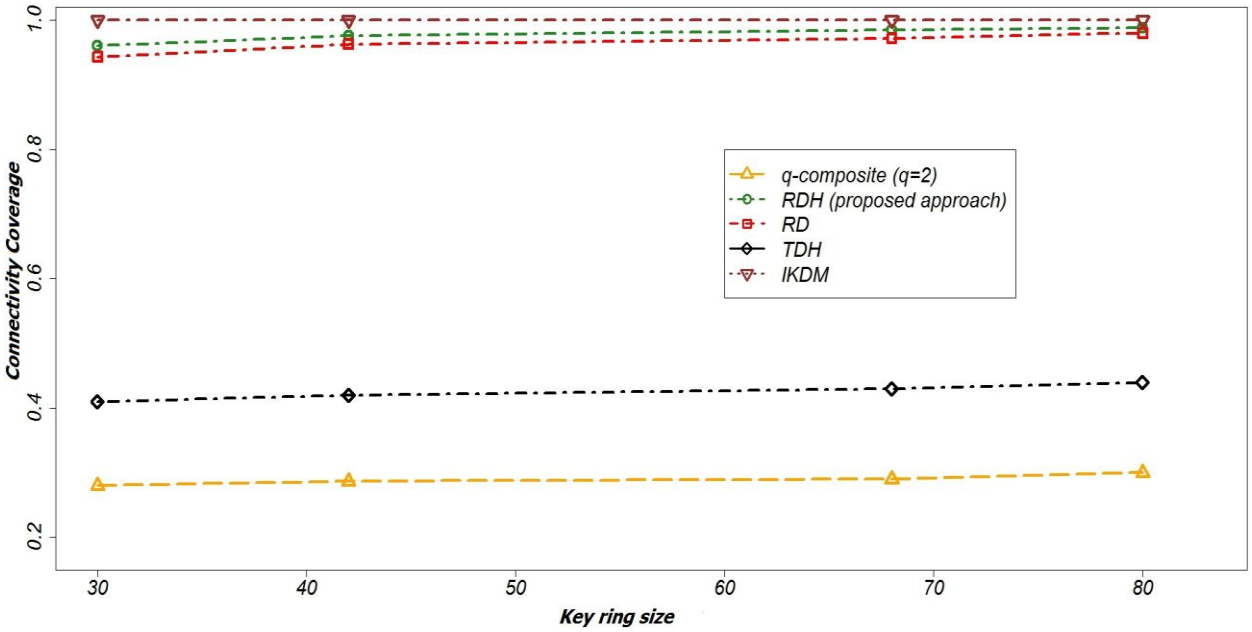

Figure 2: Connectivity Comparison. The connectivity of our scheme (RDH) is compared with classic RD, q-composite, IKDM and hierarchical TD (TDH)

common key in their radio transmission range by exchanging the list of their key identifiers. There is no eavesdropping in this concept but as will be explained in the next sections, we try to increase the resiliency against compromising nodes.

5- Analysis

In this section, a theoretical analysis of important metrics of our proposed approach including scalability, connectivity, and resilience will be presented. In our proposed approach, the Residual Design (RD) is used as a key pre-distribution scheme. After deployment, the properties of RD would help us to calculate these metrics

5-1- Scalability

In our proposed approach we have considered q^2+q+1 clusters of size q^2 in such a way that they differ from q keys. Each cluster has q^2+q sensor nodes. Therefore the maximum network size that can be supported is:

$$(q^2+q+1)(q^2+q)+(q^2+q+1)=(q^2+q+1)^2 \qquad (1)$$

5-2- Connectivity

To compute the connectivity, at first we determine the key share probability for each cluster.

According to the structure of the RD scheme, every sensor node has common keys with its corresponding cluster head node. Therefore, the probability that any sensor node and it's cluster head node have a common key is

$$P_{CHi-Si}=1$$

Every sensor node in a cluster has common keys with q^2 other nodes. Therefore the probability that any pair of sensor nodes has at least a common key is

$$P_{Si}=\frac{q^2}{q^2+q}$$

It shows that the more increase in parameter q, the more in the connectivity of the network. Therefore by increasing the q, the above limit goes to 1. In the RD scheme [11], the probability of key share between two nodes is

$$\frac{q^2}{q^2+q}*Q_{SD}+\left(\frac{q-1}{q^2+q}\times\frac{q^2+1}{q^2+q}+\frac{q^2}{q^2+q}\times\frac{q^2-q+1}{q^2+q}\right)*Q_{DC} \qquad (2)$$

where

$$Q_{SC} = \frac{\binom{q^2+q}{2}}{\binom{(q^2+q)(q^2+q+1)}{2}}$$

and

$$Q_{DC} = \frac{\binom{q^2+q}{1}\binom{q^2+q}{1}}{\binom{(q^2+q)(q^2+q+1)}{2}}$$

In the q-composite scheme [7], two nodes must share at least q common keys to be able to establish a secure link. The connectivity probability of q-composite scheme is calculated as:

$$p_c = p(q) + \ldots + p(k)$$

where

$$p(i) = \frac{\binom{p}{i}\binom{p-i}{2(k-i)}\binom{2(k-i)}{k-i}}{\binom{p}{k}^2}$$

According to the analysis in [14], the connectivity between two nodes in TDH scheme is:

$$\frac{\sum_{j=1}^{c} n_j \times Pr_j}{N}$$

where Pr_j is connectivity for each cluster, n_j is the number of sensor nodes in cluster j, and c is the number of clusters.

In [15], each pair of nodes has a unique pairwise key, therefore the network has full connectivity in IKDM.

5-3- Security Analysis

Node capture attack is a serious threat in wireless sensor networks. The resilience is the ability to resist of the network against the compromising of sensor nodes. In our proposed approach, according to the structure of cluster heads and communication between them, after compromising of any cluster head, nothing would happen for the secure communication between other cluster heads and sensor nodes. Compromising any cluster head leads to deleting the connection between sensor nodes corresponding to this cluster and base station.

To calculate the resilience, we consider the probability that a link in a cluster is compromised when an attacker capture x nodes in a cluster. This probability can be defined as

$$P(L|C_x) = \sum_{\forall j} P(l_j)P(D_j|C_x)$$

In our proposed approach, each key exists in $r = q + 1$ nodes and two communicating nodes must have a common key j in their key-rings. So the probability that a link is secured with key j is

$$P(l_j) = \frac{\binom{q+1}{2}}{\binom{q^2+q}{2} - \frac{q^2+q}{2}} = \frac{1}{q^2+q-2}$$

The probability that the key j appears in one or more of x compromised nodes is:

$$P(D_j|C_x) = 1 - \frac{\binom{q^2-1}{x}}{\binom{q^2+q}{x}}$$

Therefore, the probability that a link is compromised when x key-rings are captured by an attacker can be computed as:

$$P(L|C_x) = \frac{q^2+q}{q^2+q-2} P(D_j|C_x) \simeq P(D_j|C_x) = 1 - \frac{\binom{q^2-1}{x}}{\binom{q^2+q}{x}}.$$

The probability of compromising a secure link is computed by Camptepe and Yener in [8] for symmetric design as:

$$1 - \frac{\binom{q^2}{x}}{\binom{q^2+q+1}{x}}.$$

In [6], the authors proved that the resilience of q-composite scheme is

$$\sum_{i=q}^{k} (1 - (1 - \frac{k}{p})^x)^i \times \frac{p(i)}{p_c}.$$

In the RD scheme [11], the resilience of network against node capture is

$$(q^2+q+1)\frac{\binom{q^2(q+1)}{2}}{\binom{(q^2+q+1)(q^2+q)}{2}}\left(1 - \frac{\binom{q^4+q^3+q^2+q}{x}}{\binom{(q^2+q+1)(q^2+q)}{x}}\right). \quad (3)$$

The resistance of the network against the capture node in IKDM[15] depends on degree the of bivariate symmetric polynomials.

6- Experimental Results

In this section, we compare our work to the related works based on scalability, connectivity, and network resilience.

For numerical results, we used the C# programming language to provide a simulation tool. Then we produce our comparing graphs with the R programming language. Figure 2 shows that the connectivity estimated by our proposed approach (RDH), q-composite scheme [7], RD scheme [11], and proposed scheme in [14] (TDH) for different key-ring size. It can be observed that our proposed approach has better connectivity than the other related

schemes.

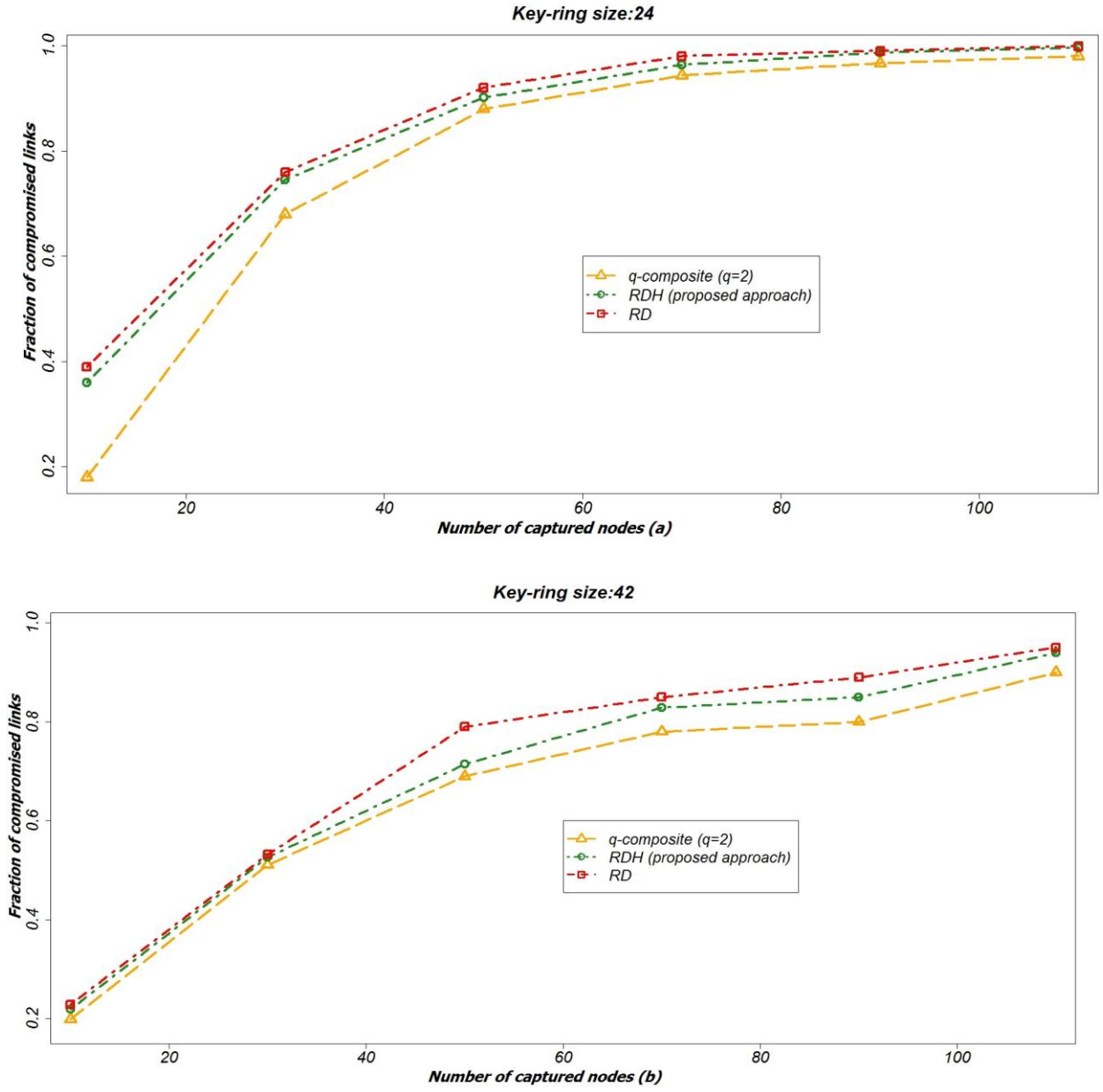

Figure 3: Security behavior Comparison. The network resilience of our new approach (RDH) is compared with classic residual resign (RD) and q-composite design. Simulation results are computed for the same key-ring size k = 24 in (a) and k = 42 in (b)

It shows that by increasing the key-ring size our proposed approach has full connectivity. Figure 3 compares the resilience against node capture attack for key-ring sizes $k = 24$ and $k = 42$. The figure shows that for an equal key ring size, the resilience of our proposed approach is better than the RD scheme. It also has the same resilience with the q-composite scheme for compromised nodes numbers greater than 100.

7- Conclusion

In this paper, we provide a novel proper key distribution approach for large-scale clustered hierarchical wireless sensor networks based on Residual Design. We show that the proposed architecture in wireless sensor networks is more convenient for hierarchical scalable network forms. The analysis and numerical results show that our proposed RDH design has the best connectivity coverage and also scalability rather than other similar schemes. Also this approach provides good security against all types of sensor nodes capture no matter if the cluster head or sensor node is compromised.

Our future work would target to improve low resilience against some other types of attacks. We will also focus on the models based on the Internet of Things (IoT) in this idea.

References

[1] C. Boyd, A. Mathuria, and D. Stebila, Protocols for authentication and key establishment. Springer, 2003.

[2] I. Memon, "A secure and efficient communication scheme with authenticated key establishment protocol for road networks," Wireless Personal Communications, vol. 85, no. 3, pp. 1167-1191, 2015.

[3] I. Memon, I. Hussain, R. Akhtar, and G. Chen, "Enhanced privacy and authentication: An efficient and secure anonymous communication for location based service using asymmetric cryptography scheme," Wireless Personal Communications, vol. 84, no. 2, pp. 1487-1508, 2015.

[4] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," Journal of network and computer applications, vol. 33, no. 2, pp. 63-75, 2010.

[5] J.-W. Dong, D.-Y. Pei, and X.-L. Wang, "A class of key predistribution schemes based on orthogonal arrays," Journal of Computer Science and Technology, vol. 23, no. 5, pp. 825-831, 2008.

[6] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in Proceedings of the 9th ACM conference on Computer and communications security, 2002, pp. 41-47.

[7] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in 2003 Symposium on Security and Privacy, 2003, 2003: IEEE, pp. 197-213.

[8] S. A. Çamtepe and B. Yener, "Combinatorial design of key distribution mechanisms for wireless sensor networks," IEEE/ACM Transactions on networking, vol. 15, no. 2, pp. 346-358, 2007.

[9] J. Lee and D. R. Stinson, "On the construction of practical key predistribution schemes for distributed sensor networks using combinatorial designs," ACM Transactions on Information and System Security (TISSEC), vol. 11, no. 2, pp. 1-35, 2008.

[10] X. Zhang and J. Wang, "An efficient key management scheme in hierarchical wireless sensor networks," in 2015 International Conference on Computing, Communication and Security (ICCCS), 2015: IEEE, pp. 1-7.

[11] V. Modiri, H. H. S. Javadi, and M. Anzani, "A novel scalable key pre-distribution scheme for wireless sensor networks based on residual design," Wireless Personal Communications, vol. 96, no. 2, pp. 2821-2841, 2017.

[12] M. Anzani, H. H. S. Javadi, and V. Modirir, "Key-management scheme for wireless sensor networks based on merging blocks of symmetric design," Wireless Networks, vol. 24, no. 8, pp. 2867-2879, 2018.

[13] S. Ruj and B. Roy, "Key pre-distribution using partially balanced designs in wireless sensor networks," International Journal of High Performance Computing and Networking, vol. 7, no. 1, pp. 19-28, 2011.

[14] M. Javanbakht, H. Erfani, H. H. S. Javadi, and P. Daneshjoo, "Key predistribution scheme for clustered hierarchical wireless sensor networks based on combinatorial designs," Security and Communication Networks, vol. 7, no. 11, pp. 2003-2014, 2014.

[15] Y. Cheng and D. P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," Ad Hoc Networks, vol. 5, no. 1, pp. 35-48, 2007.

[16] Y. Zhang and P. Li, "Key management scheme based on nodes capture probability for wireless sensor networks," in 2018 Chinese Control and Decision Conference (CCDC), 2018: IEEE, pp. 5470-5475.

[17] A. Albakri, L. Harn, and S. Song, "Hierarchical key management scheme with probabilistic security in a wireless sensor network (WSN)," Security and Communication Networks, vol. 2019, 2019.

[18] D. Stinson, Combinatorial designs: constructions and analysis. Springer Science & Business Media, 2007.

[19] T. Kavitha and R. Kaliyaperumal, "Energy Efficient Hierarchical Key Management Protocol," in 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019: IEEE, pp. 53-60.

[20] S. Zhao, K. Tepe, I. Seskar, and D. Raychaudhuri, "Routing protocols for self-organizing hierarchical ad-hoc wireless networks," 2003.

[21] B. Liu, Z. Liu, and D. Towsley, "On the capacity of hybrid wireless networks," in IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428), 2003, vol. 2: IEEE, pp. 1543-1552.

[22] G. Gupta and M. Younis, "Performance evaluation of load-balanced clustering of wireless sensor networks," in 10th International Conference on Telecommunications, 2003. ICT 2003, 2003, vol. 2: IEEE, pp. 1577-1583.
[23] J. Li and R. Levy, "Fair and Secure Clustering Scheme (FSCS) clustering protocol," In Technical report, 2005.
[24] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," IEEE Transactions on mobile computing, vol. 3, no. 4, pp. 366-379, 2004.

Vahid Modiri received the Ph.D. degree in Computer Science at the Department of Computer Engineering, Islamic Azad University, Science and Research. He received the B.S. degree in Computer Science at Iran University of Science and Technology and the M.S. degree in Computer Science from the Islamic Azad University North Tehran branch. His main research focuses on key management in wireless sensor networks and wireless security.

Hamid Haj Seyyed Javadi received the B.S., M.S. and Ph.D. degrees in Amir Kabir University. He has been working as a full-time faculty member and Associate Professor of Shahed University. His research interests are ad-hoc network technologies, sensor network technology, distributed operating systems, and heuristic Algorithm.

Amir Masoud Rahmani was born in 1974 in Iran. He received his B.S. in computer engineering from Amir Kabir University, Tehran, in 1996, the M.S. in computer engineering from Sharif University of technology, Tehran, in 1998 and the PhD degree in computer engineering from Islamic Azad University (IAU), science and research branch, Tehran, in 2005. His research interests are in the areas of distributed systems, ad hoc and wireless sensor networks and evolutionary computing.

Mohaddese Anzani received the Ph.D. degree in Computer Science at Shahed University, Tehran, Iran. She received her M.S. degree in Mathematics from Shahed university, Tehran, Iran. Her research interests include key management in wireless sensor networks and wireless security.