2020, Volume 7, Number 1 (pp. 81–94)
http://www.jcomsec.org

Journal of Computing and Security

Mining Association Rules from Semantic Web Data without User Intervention

Reza Ramezani[a,*], Mohammad Ali Nematbakhsh[a], Mohamad Saraee[b]

[a] *Department of Software Engineering, Faculty of Computer Engineering, University of Isfahan, Iran.*
[b] *School of Computing, Science and Engineering, University of Salford, Manchester, UK.*

ARTICLE INFO.

Article history:
Received: 22 February 2020
Revised: 15 May 2020
Accepted: 7 June 2020
Published Online: 19 July 2020

Keywords:
Semantic Web, Association Rules, Data Mining, SWApriori.

ABSTRACT

With the introduction and standardization of the semantic web as the third generation of the web, this technology has attracted and received more human attention than ever. Thus, the amount of semantic web data is continuously growing, which makes them a rich source of useful data for data mining techniques. Semantic web data have some complexities, such as the heterogeneous structure of data, the lack of well-defined transactions, and the existence of typed relations between items. In this paper, a new technique named *SWApriori* is presented, which by using both entities and relations in the extraction of frequent itemsets, generates a new class of association rules (ARs) from semantic web data. The proposed technique by considering the complex heterogeneous nature of semantic web data, without any need to a domain expert, and without any data conversion to transactional data format extracts ARs from semantic web data directly. For evaluation, the proposed technique is applied to *Factbook* and *DBPedia* datasets. The experimental results demonstrate the ability of the proposed technique in extracting relational ARs from semantic web data by considering the mentioned challenges. Supplementary experiments show that the proposed technique can extract interesting patterns that are not discoverable by state-of-the-art association rule mining techniques.

1 Introduction

Since the advent of RDF[1], RDFS[2], and OWL[3] standardization, people have a better understanding of the semantic web. Thus, the amount of semantic web data is continuously growing [1]. With the increasing data publishing from different sources, there is now a large amount of semantic web data. The ontological metadata and the meaningful entities and relations in semantic web data improve the effectiveness of data mining techniques and cause them to discover richer and more useful knowledge [2].

Semantic web data mining has attracted much attention in recent years. Association rule mining (ARM) is one of the most important data mining techniques which aims at finding frequent itemsets and association rules (ARs). Most of the existing ARM techniques

* Corresponding author.

Email addresses: `r.ramezani@eng.ui.ac.ir` (R. Ramezani), `nematbakhsh@eng.ui.ac.ir` (M. A. Nematbakhsh), `m.saraee@salford.ac.uk` (M. Saraee)

https://dx.doi.org/10.22108/jcs.2020.121763.1046
ISSN: 2322-4460 © 2020 JComSec. All rights reserved.

[1] http://www.w3.org/TR/rdf-concepts/
[2] http://www.w3.org/TR/rdf-schema/
[3] http://www.w3.org/TR/owl-features/

deal with transactional data in a tabular format [3] or a graph-based structure [4]. Compared to the traditional data, there exist some difficulties in semantic web data, as:

- The absence of the definition of transactions in semantic web data
- The heterogeneous structure of semantic web data
- The existence of different relations between entities
- The need for domain expert and the end-user intervention in the mining process

Most of the existing semantic web data mining approaches convert the semantic web data to traditional ones and then apply the data mining techniques. Such a conversion leads to losing some interesting knowledge and patterns. In this paper, a new technique named Semantic Web Apriori *(SWApriori)* has been presented to address the problem of ARM from semantic web data without any data conversion and by considering the aforementioned challenges. To overcome the problems of **no exact definition of transactions** and **need for a domain expert**, the proposed technique uses a novel approach in ARM by which large itemsets are generated directly from semantic web datasets without any need for transactions and the end-user intervention. To deal with the **heterogeneous data structure**, a linked list-based data structure is proposed and used within the mining process. Finally, to take **different relations between entities** into account, a new concept of Item is proposed, which is composed of one Entity and one Relation. From the semantic web point of view, an Entity is the object of a statement and the Relation is its incoming predicate.

The performance of the proposed technique has been evaluated by conducting several experiments on semantic web datasets. In this regard, the ability of the proposed technique in generating relational ARs from semantic web data by considering the above challenges has been demonstrated by an illustrative example and experimental evaluations on *Factbook* and *DBPedia* datasets. Then, several interesting generated ARs and the computation time of the proposed technique have been presented. Finally, supplementary experiments have been conducted to compare the proposed technique with existing ARM techniques. The results show that the *SWApriori* technique can generate interesting relational ARs that state-of-the-art techniques cannot generate them.

The rest of the paper is organized as follows. Section 2 introduces several related works. Section 3 contains the general methodology and foundations of the proposed technique, shows an illustrative example, and then presents the pseudo-code of the proposed algorithm. Section 4 gives the experimental results. Finally, Section 5 concludes the paper and offers suggestions for future work.

2 RELATED WORK

In the past, many machine learning algorithms have been successfully applied to traditional datasets to discover useful and previously unknown knowledge. Although these algorithms are useful, the nature of semantic web data is quite different from the traditional ones [5]. Most ARM algorithms deal with traditional datasets [6, 7] which can be classified into two main categories: Apriori-inspired techniques and FP-Tree based techniques.

The semantic web data mining has applications in different domains, such as healthcare systems [8], recommender systems [9], education [10], and market basket analysis [11]. These techniques first require to refine data [12] to generate interesting patterns from RDF data [5].

Some ARM techniques are based on inductive logic programming [13], which uses the ontology encoded logic. For example, the technique proposed in [14] uses a declarative language to extract horn-typed ARs among relations on small sets of conjunctive queries. This technique has then been extended to speed up the mining process [15]. A framework has been proposed by [16] for mining and classifying generalized rules at different levels of abstraction. Similar tools have been developed for mining semantic web data which convert these data to traditional ones to mine ARs using Apriori and FP-Growth algorithms [17].

Another approach in ARM is the use of frequent sub-graph and sub-tree techniques for pattern discovery from graph-structured data [18]. These algorithms are based upon mining a tree/graph generated from existing transactions [19]. Although these techniques are interesting, since there is no exact definition of transactions in semantic web data, they are not appropriate for the semantic web.

All of the above studies deal with traditional data. Therefore, some data transformations are required to make the semantic web data suitable for these algorithms. In this regard, a new technique has been presented in [20] that by using a mining pattern, provided by the end-user, converts semantic web data to traditional data in which transactions consist of the values of entities. Then, traditional ARM algorithms are employed to generate ARs. In this approach, to provide a mining pattern and convert semantic web data, the end-user should have deep knowledge about the structure of the dataset and the ontology that sustains it. Similarly, a user-centric pattern mining

Table 1. Combinations of Triple Parts [24].

	Context	Target	Use Case
1	Subject	Predicate	Schema discovery
2	Subject	Object	Basket analysis
3	Predicate	Subject	Clustering
4	Predicate	Object	Range discovery
5	Object	Subject	Topical clustering
6	Object	Predicate	Schema matching

approach has been proposed in [21]. To cope with the transaction definition problem, the authors of [22] have suggested that besides the application ontology, a new ontology is generated to define the specifications of the concepts, such as item, transaction, and their properties to be used in the ARM process.

A new class of ARs named *Multi-Relation* ARs has been proposed in [23] which can be extracted from semantic web data and relational databases. In contrast to primitive, simple, and even multi-relational ARs (that are usually extracted from multi-relational databases), the rule items of this kind of ARs consist of one item but several relations. These relations indicate the indirect relationship between items.

In RDF, each data statement names a *Triple* and has three parts: *subject*, *predicate*, and *object*. As there is no exact definition of transactions in semantic web data, to generate transactions, the technique presented in [24] has used one of these three parts to group transactions (TID) and one of the remaining ones as transaction items. Different combinations of transaction ID and transaction items lead to various ARs with different usages. Six different combinations of these parts, along with their usage, are shown in Table 1. For example, grouping triples by the predicate and using subjects for generating transactions has usage in clustering items based on the similarity of their behavior. This approach eliminates one part of triples and does not consider it in the mining process, which, as a result, causes losing some knowledge. In a similar approach, an Apriori algorithm has been used by [25] to generate ARs with two items from semantic web data in the healthcare domain. In this work, each patient is seen as a transaction, but the relations between description items of the patients are ignored by the underlying algorithm. The work presented by [26] proposes a new type of transaction by adding a timestamp to triples for constructing sequential semantic transactions suitable for stock market prediction.

In addition to the introduced techniques, several tools, such as LiDDM [27], have been presented to facilitate data mining processes (clustering, classification, and ARs) over semantic web data [28]. Similar to other techniques, LiDDM converts semantic web data to traditional ones in a tabular format and then employs traditional data mining algorithms. A centralized approach to extract ARs from linked data has been presented in [29] which collects the required data from multiple data sources. RapidMiner semweb plugin [30] is a similar approach to mining semantic web data and supports reformatting set-valued data. As with LiDDM, in the RapidMiner, the end-user has to define a suitable SPARQL query for retrieving interested data. SPARQL-ML [31] is another work that provides a particular statement as an extension to SPARQL query language to create and learn a model for a specific concept of the retrieved data. In all these techniques and tools, to convert the semantic web data to the traditional ones, the end-user should have deep knowledge about the structure of the dataset and its concerned ontology.

In this paper, a new ARM algorithm has been presented that without converting any semantic web data and the end-user intervention, generates ARs from the semantic web datasets directly.

3 MOTIVATION AND METHODOLOGY

In this section, an illustrative example is provided to present a detailed view of the presented technique, along with the concerned definitions, step by step. Then, the underlying data structure and the proposed algorithm will be described in detail.

3.1 Challenges and Problems

There exist several challenges in extracting ARs from semantic web data, as follows:

1- **No exact definition of transactions in semantic web data**: In conventional transactional systems, the structure of transactions is already defined, and they are stored in datasets with predefined schemas. The transactions can then be queried from these datasets using well-defined structures. In this regard, a transaction can be considered as a set of items that have a specific relation (such as *being together*) with each other [32]. In contrast, there is no exact definition of transactions in semantic web data [22]. In semantic web data, data are heterogeneous and there exist different relations between items.

Thus, since the definition of transactions is not intuitive in semantic web data, the way of constructing transactions from these data is a challenge. There are two approaches to address it. The

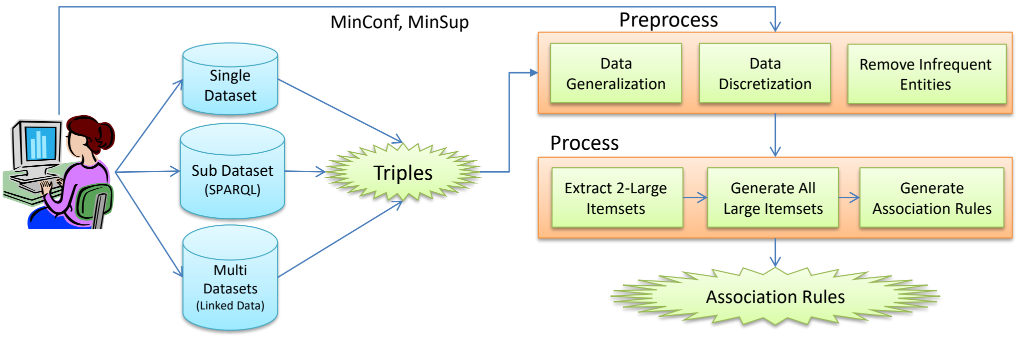

Figure 1. Workflow of the Proposed Solution.

first one, similar to the existing approaches, requires proposing a new concept of transactions in semantic web datasets by involving the end-user. The second one is to propose a new algorithm that does not deal with transactions in the ARM process. Our proposed technique follows the latter.

2- **The heterogeneous structure of semantic web data**: Traditional data mining algorithms work with homogeneous datasets in which the data instances have predefined attributes. In contrast, the semantic web data model leads to heterogenic data [33]. This means that particular category/domain instances (such as countries, cars, drugs, etc.) might have different attributes based on one or multiple ontologies. The proposed algorithm uses a linked list-based data structure to deal with these heterogeneous data in the mining process.

3- **The existence of different relations between items**: Most traditional ARM algorithms, to generate large itemsets, use only the value of the items, and they assume there exists only one type of relations between items (for example, *bought together*). In contrast, in semantic web data, there are different relations or typed links (a.k.a. predicates) between entities. Our proposed algorithm considers all relations of the entities in the generation of frequent itemsets. Thus, in the presented approach, an *Item* not only is an *Entity* but also it consists of one *Entity* and one *Relation*.

4- **The need for a domain expert and the end-user intervention in the mining process**: As mentioned before, in the existing semantic web data mining techniques, the end-user should be aware of the dataset and ontology structure to generate transactions [20, 22, 27]. In this study, a new algorithm is proposed that does not involve the end-user within the mining process.

Our proposed solution addresses all the above challenges and uses an Apriori-inspired approach to generate large itemsets and ARs. In contrast to the traditional Apriori algorithm, the proposed technique does not depend on the concept of a transaction. This technique generates **2-large itemsets** from the semantic web data, without taking transactions into an account. Afterward, **larger itemsets** are generated from these 2-large itemsets. The generated itemsets differ from the traditional ones in the sense that their items consist of two parts: *Entity* and *Relation*. In this context, an *Entity* is an object and *Relation* is its incoming predicate. Finally, the **association rules** are generated from the large itemsets.

Figure 1 shows the workflow of the proposed technique in which a dataset is first provided. Then, in the preprocessing step, the entities are generalized and discretized, and the infrequent ones are eliminated. Finally, ARs are generated in the process step, as described above. This workflow will be elaborated more in the next sections.

3.2 An Illustrative Example

A) *Sample Dataset*

To better clarify the proposed technique, an illustrative example is presented in this section. Some triples describing people in the university domain are tabulated in Table 2. The description of the data items is presented at the bottom of the table. Figure 2 shows the information of Table 2 in a different format. This format is used by the proposed technique.

B) *2-Large Itemset*

In the proposed technique, after preprocessing and elicitation of the input data using the underlying ontology, the first step of mining ARs from semantic web data is the generation of 2-large itemsets. To generate these itemsets, the triples are grouped first by objects values, and then for each group of object values, the subjects are grouped by their outgoing predicates values. The result of such a grouping the triples of Table 2 is depicted in Figure 2. As this figure shows, the subjects of triples that have a correspondingly equivalent predicate and object values constitute

Table 2. Input Dataset Contents (Example).

Subject	Predicate	Object
Mohammed	Supervised by	David
Mohammed	Supervised by	Jack
Mohammed	Marital Status	Bachelor
Mohammed	Student at	USC
Mohammed	Knows	Jack
Mohammed	Knows	Daniel
Mohammed	Knows	Harry
Mohammed	Degree	M.Sc.
Harry	Supervised by	Joseph
Harry	Marital Status	Bachelor
Harry	Student at	USC
Harry	Degree	M.Sc.
Harry	Knows	Jack
Harry	Friend with	Mohammed
Harry	Friend with	Daniel
Daniel	Marital Status	Bachelor
Daniel	Student at	MIT
Daniel	Friend with	Mohammed
Daniel	Knows	Jack
Daniel	Degree	M.Sc.
Oliver	Supervised by	David
Oliver	Marital Status	Married
Oliver	Student at	USC
Oliver	Degree	Ph.D.
David	Marital Status	Married
David	Teach in	USC
David	Knows	Mohammed
David	Knows	Oliver
David	Degree	Ph.D.
Jack	Friend with	David
Jack	Marital Status	Married
Jack	Teach in	MIT
Jack	Knows	Mohammed
Jack	Degree	Ph.D.
Joseph	Teach in	USC
Joseph	Marital Status	Married
Joseph	Degree	Ph.D.
Description: David, Jack, and Joseph are professors. USC (University of Southern California) and MIT (Massachusetts Institute of Technology) are universities. M.Sc. and Ph.D. are educational degrees.		

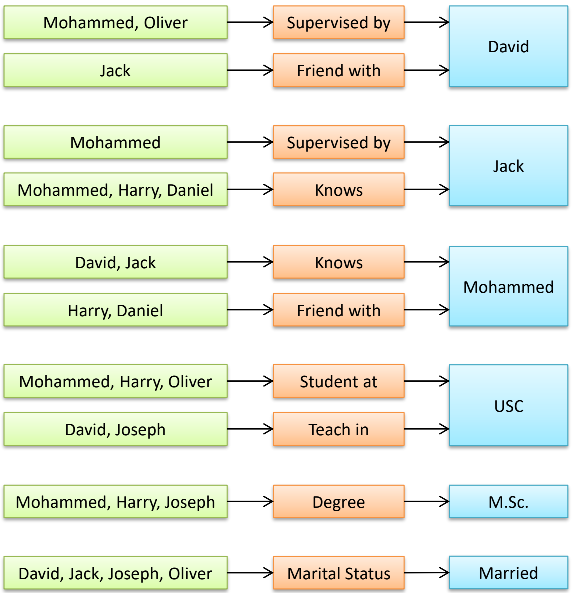

Figure 2. Grouping the Triples of Table 2.

a subject set.

After grouping triples, the proposed algorithm compares the subject sets of different groups two by two. If the number of common subjects of the two subject sets is equal to or greater than a predefined minimum support value (*MinSup*), the corresponding object and predicate values of the two subject sets make a 2-large itemset. In the generated itemsets, each item consists of the values of both predicates and objects.

As an example, in Figure 2, *Jack* is an object and *Knows* is one of its incoming relations (predicates) which have *Mohammed*, *Harry*, and *Daniel* as their subject set. Similarly, *USC* is an object and *Student* at is one of its incoming relations which have *Mohammed*, *Harry*, and *Oliver* as their subject set. Now, suppose that the algorithm compares the subject sets of (*Knows* + *Jack*) with that of (*Student at* + *USC*). Intersection of (*Mohammed*, *Harry*, *Daniel*) and (*Mohammed*, *Harry*, *Oliver*) yields (*Mohammed*, *Harry*) as common subjects. If 2 -the number of common subjects- is equal to or greater than *MinSup* value, (*Knows* + *Jack*), (*Student at* + *USC*) will generate a 2-large itemset. This 2-large itemset means that the *subjects* that are *Student at University of Southern California (USC)*, they also *Know* Dr. Jack.

As another example, (*Knows* + *Jack*), (*Degree* + *M.Sc.*) can generate a 2-large itemset, because they have two common subjects as *Mohammed*, *Harry*. Thus, we will have

- (*Knows* + *Jack*), (*Student at* + *USC*)
- (*Knows* + *Jack*), (*Degree* + *M.Sc.*)

as 2-large itemsets. After generating 2-large itemsets, the proposed technique will generate larger itemsets.

C) *Larger Itemsets*

The traditional Apriori algorithm, to generate an (L+1)-large itemset, combines two L-large itemsets whose L-1 first items are equal. The obtained itemset is large if its occurrence is equal to or greater than the predefined *MinSup* value, and also all its subsets are large as well. In our proposed technique, the (L+1)-large itemsets are generated by first combining the two L-large itemsets whose L-1 first items have an equal entity and relation values. Then, an intersection is taken from the subject sets of all L+1 items of the generated (L+1)-itemset to assess whether the itemset is large or not. With this strategy, the proposed algorithm will consider both entities and relations in generating large itemsets.

In the aforementioned example, since the first item of both 2-large itemsets is equal to (*Knows* + *Jack*), their combination generates an itemset with three items, as (*Knows* + *Jack*), (*Student at* + *USC*), (*Degree* + *M.Sc.*). If the number of common subjects of subject sets of these three items is equal to or greater than the MinSup value, this itemset will be a 3-large itemset, as:

- (*Knows* + *Jack*), (*Student at* + *USC*), (*Degree* + *M.Sc.*)

In this example, (*Mohammed*, *Harry*) is the common subject set of the obtained 3-large itemset. It can be seen from Figure 2 that all subsets of the obtained 3-large itemset are large as well. Generating larger itemsets will continue until it is no longer possible to generate new itemsets.

D) *Association Rules* Finally, ARs are generated using large itemsets. To have simple yet useful rules, the proposed technique generates ARs with only one item in the consequent part. In this context, ARs with a confidence equal to or greater than the predetermined minimum confidence value (*MinConf*) are marked as strong rules. The confidence of an AR is calculated as the support of the items that appear in the entire rule divided by the support of the items that appear in the antecedent part.

In the above example (*Knows* + *Jack*), (*Student at* + *USC*), (*Degree* + *M.Sc.*) is a large itemset. Since it has three items, three different ARs can be generated, as:

- **Student at** (*USC*), **Knows** (*Jack*) → **Degree** (*M.Sc.*)
- **Knows** (*Jack*), **KDegree** (*M.Sc.*) → **Student at** (*USC*)
- **Student at** (*USC*), **Degree** (*M.Sc.*) → **Knows** (*Jack*)

The first rule means that many *USC* students that

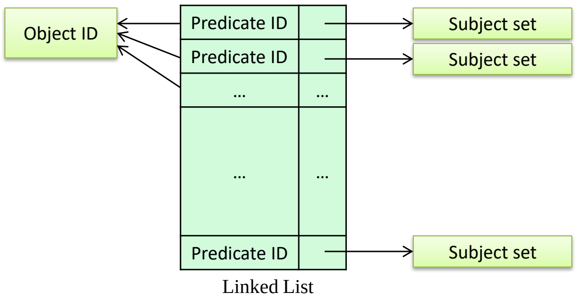

Figure 3. ObjectInfo Structure.

know *Jack* too, with a certain probability (rule confidence), their educational degree is *M.Sc.*

The presented example shows that despite there were no transactions in the provided example, the proposed technique discovered ARs from the presented semantic web data directly, without converting these data to traditional ones.

3.3 Data Structure

To deal with the heterogeneous structure of the semantic web data, a linked list-based data structure is used to store and organize the input data to be processed by the proposed algorithm. To do so, an ObjectInfo class with the following attributes has been defined, as:

1- ***Object ID***: To store the value of an object
2- A ***Linked List*** whose entries have two parts:
   a. ***Predicate ID***: To store the values of incoming predicates of the object
   b. ***Subjects Set***: pointer to a subject set. The subject set contains the subjects which refer to the *Object ID* via the corresponding *Predicate ID* of the linked list.

The **ObjectInfo** class is depicted in Figure 3 (see Figure 2 for clarification as well). A unique instance of the **ObjectInfo** class is instantiated for each distinct object of the input dataset.

3.4 SWApriori Algorithm

The proposed algorithm is named Semantic Web Apriori (*SWApriori*) whose pseudo-code is shown in Algorithm 1. This algorithm receives as input a semantic web dataset containing triples along with its concerned ontologies, the minimum support value (*MinSup*) to extract large itemsets, and the minimum confidence value (*MinConf*) to generate strong ARs. The end-user just has to provide a semantic web dataset as well as the *MinSup* and *MinConf* values. However, he/she can also provide a more specific dataset by employing the knowledge encoded in the underlying ontology and also by using some SPARQL commands (such as FILTER) to obtain more specific ARs.

In the proposed algorithm, a preprocessing is done in line 2 to eliminate infrequent entities. At this phase, the end-user can use data generalization concepts to generalize or group the existing entities and relations so that their frequency is increased. This can be done by using some ontological properties such as *rdfs:subClassOf*, *rdfs:subPropertyOf*, *owl:equivalentClass*, *owl:equivalentProperty*, and *owl:sameAs*. For example, the strategy presented by [34] uses *rdf:type* and *rdfs:subClassOf* relations in the ontology to boost the semantics of the input data. Besides, since data discretization is done in this step, some properties such as *rdfs:Datatype*, *rdfs:range*, *rdfs:domain*, *owl:allValuesFrom*, and owl:someValuesFrom can be used for the data discretization by taking the underlying constraints into account. Finally, at this step, the manipulated data are stored in appropriate instances of the *ObjectInfo* class.

The **Generate2LargeItemsets** function is invoked in line 3 to generate 2-large itemsets. The workflow of this function was previously described in Section 3.2. The complexity of this function is in the order of O($L^2R^2S^2$), where L is the number of large entities (*ObjectInfo instances*), R is the maximum number of relations in *ObjectInfo* instances, and S is the maximum length of subject sets in the *ObjectInfo* instances.

The loop of lines 5 to 22 generates L-large itemsets ($L\geq3$). This loop iterates for different values of L, and in each iteration, all L-large itemsets are compared two by two to generate (L+1)-candidate itemsets, using **CombineAndSort** function (lines 9-13). This function sorts the items of the new candidate itemset first by *Entity* value and then by *Relation* value. The new candidate itemset is then added to the candidate itemsets collection (line 12). Each loop iteration (lines 8-14) uses the large itemsets generated by the previous loop iteration which had been stored in *LIs*. After generating all candidate itemsets with the length of L+1, the (L+1)-large itemsets are selected in lines 16 to 30 to be added to the collection of all large itemsets (*AllLIs*) in line 21. The worst-case complexity of this function has the order of O(LS^2I^3), where L is the maximum length of large itemsets, S is the maximum number of large itemsets at individual iterations, and I is the maximum number of items in large itemsets.

After generating all possible large itemsets, ARs are generated by calling the **GenerateRules function** in line 23. The time complexity of this function is in the order of *O(LI)*, where L is the number of all large itemsets and I is the maximum number of items in the large itemsets.

Algorithm 1 SWApriori: Mining Association Rules From Semantic Web Data.

```
INPUT:
DS ← A dataset consisting of triples (Subject, Predicate, Object) and its concerned ontologies
MinConf ← Minimum support value
MinConf ← Minimum confidence value
OUTPUT:
AllLIs ← Large itemsets
Rules ← Association rules
Variables:
$LIs^4$, Candidates ← List of Itemsets
$IS$, $IS_1$, $IS_2$, $IS_3$ ← Itemset (multiple items)
Begin
Preprocess DS and store its data in the ObjectInfo instances
LIs = AllLIs = Generate2LargeItemsets(ObjectInfo[], MinSup)
L = 1
Do
L = L + 1
Candidates = null;
for each $IS_1, IS_2$ in $LIs$ do
    if $IS_1$[1..L−1]. Object = $IS_2$[1..L−1]. Object and $IS_1$[1.. L-1 ]. Predicate = $IS_2$[1.. L-1 ]. Predicate then
        $IS_3$ = CombineAndSort($IS_1, IS_2$)
        Candidates = Candidates ∪ $IS_3$
    end if
end for
  $LIs$ = null
for each $IS$ in Candidates do
    if Support($IS$) ≥ $MinSup$ AND all subsets of $IS$ are large then
        $LIs = LIs \cup IS$
    end if
end for
$AllLIs = AllLIs \cup LIS$
while $LIs$.Length > 0 do
    Rules = GenerateRules ($AllLIs, MinConf$)
end while
Return $AllLIs, Rules$
End
```

4 EXPERIMENTS AND RESULTS

4.1 Dataset

To evaluate the usefulness of the proposed algorithm, several experiments have been conducted on *Factbook* and *DBPedia* datasets. *Factbook* is a dataset in countries domain and describes different features of countries by numeric values. For the experiments, these values have been categorized into three groups: *decrease*, *normal*, and *increase*. In this paper, the entire *Factbook* dataset has been downloaded to be fed to the proposed algorithm. After preprocessing this dataset, the following information has been obtained:

- *Total Triples: 38,736*
- *Total Distinct Subjects: 255*
- *Total Distinct Predicates: 298*
- *Total Distinct Objects: 1,897*
- *The average number of outgoing Predicates for each Subject: 151.9*
- *The average number of incoming Predicates for each Object: 20.41*

4.2 Experimental Results

In the first evaluation, several experiments have been conducted to reveal the effects of applying the *SWApriori* algorithm with different *MinSup* values to the *Factbook* dataset. In these experiments, the *MinConf* value is set as 0.7 and the *MinSup* value range is set between 0.54 and 0.8. Introducing a dataset-independent criterion for choosing appropriate values for these parameters is out of the scope of this paper. Thus, for evaluation purposes, they have been selected experimentally. With the *Factbook* dataset, *MinSup* values less than 0.54 cause to generate a large number of

Table 3. Some Discovered ARs Along With Their Confidence and Support Values.

Rule	**Confidence**	**Support**
Investment_GrossFixed(increase) → Debt_External(decrease)	97%	61.2%
Investment_GrossFixed(decrease) → GDP_OfficialExchangeRate(decrease)	92%	72.2%
GDP_OfficialExchangeRate(increase) → UnemploymentRate(decrease)	98%	61.4%
InflationRate_ConsumerPrices(increase) → GDP_OfficialExchangeRate(decrease)	90%	71.5%
InfantMortalityRate_Total(increase) → UnemploymentRate(decrease)	98%	57%
InfantMortalityRate_Total(increase) → Literacy_TotalPopulation(increase)	98%	57.4%
Literac_TotalPopulation(increase) → UnemploymentRate(decrease)	94%	67.7%
InflationRate_ConsumerPrices(decrease) → UnemploymentRate(decrease)	96%	57%
MilitaryExpenditures_DollarFigure(decrease) → Oil_Imports(decrease)	91%	77.7%
InflationRate_ConsumerPrices(decrease) → Oil_Imports(decrease)	96%	59.9%
Oil_ProvedReserves(decrease) → IndustrialProductionGrowthRate(decrease)	95%	61.2%
Imports(decrease) → UnemploymentRate(decrease)	94%	64.5%
Literacy_TotalPopulation(increase) → Imports(decrease)	97%	56.2%
GovernmentType(republic) → Suffrage(18)	97%	56.8%
InflationRate_ConsumerPrices(decrease) UnemploymentRate(decrease) → Imports(decrease)	94%	63.3%
GDP_OfficialExchangeRate(decrease) Investment_GrossFixed(increase) → Debt_External(decrease)	95%	59.5%
GDP_OfficialExchangeRate(decrease) Debt_External(decrease) → GDP_PurchasingPowerParity(decrease)	93%	76.1%
UnemploymentRate(increase) Debt_External(decrease) → GDP_OfficialExchangeRate(decrease)	95%	58.6%

ARs, whereas no ARs were generated with *MinSup* values more than 0.8.

Table 3 shows several ARs generated by the proposed *SWApriori* algorithm along with their confidence and support values. In each rule, the first sentence indicates a predicate, and the inter parenthesis value indicates an object. For example, the first rule in the table indicates that by increasing the gross investment of a country, its external debt is decreased with a 97% confidence. The experiments and the obtained results demonstrate the ability of the proposed algorithm in mining ARs from semantic web data directly, without the end-user intervention and semantic web data conversion.

For different *MinSup* values, Table 4 shows the algorithm behavior from different points of view. In this table, the second column (*1-Item*) shows the number of objects marked as large entities. In other words, it indicates how many *ObjectInfo* instances have been generated. As mentioned before, the number of *ObjectInfo* instances has an L^2 effect on the complexity of the **Generate2LargeItemsets** function.

- The third column ($2 - Item$) shows the number of 2-large itemsets generated by **Generate2LargeItemsets** function. These itemsets are then used by the *SW Apriori* algorithm to generate larger itemsets. The number of 2-large itemsets has an S^2 effect on the time complexity of generating large itemsets. In the worst case, the number of 2-large itemsets equals L^2R^2, whereL is the number of ObjectInfo instances and R is the maximum number of relations of $ObjectInfos$.
- The fourth column (*Large Itemsets*) shows the number of large itemsets generated for different *MinSup* values. As the obtained results show, the number of generated large itemsets depends on the number of 2-large itemsets. In addition, it has an S effect on both the time complexity of the **GenerateRules** function and the number of generated rules.
- The fifth column (*Rule Count*) indicates the number of generated strong rules and reveals how this number depends on the *MinSup* value and the number of large itemsets.
- The sixth column (*Confidence*) shows the average confidence value of the generated rules. In this experiment, the average confidence value is between 0.921 and 0.96.
- Finally, the last column (*Execution Time*) tabulates the execution time of the proposed algorithm for different *MinSup* values. The proposed algorithm has been implemented with C# 4.5

Table 4. Statistical Results of Mining ARs From Factbook.

MinSup	1-Item	2-Item	Large Itemsets	Rules Count	Confidence	Execution Time
0.54	59	1170	228943	1315999	0.96	1203 sec
0.56	58	1011	86154	456429	0.959	1258 sec
0.58	53	870	32963	159428	0.957	1047 sec
0.6	50	709	12966	57352	0.957	861 sec
0.62	48	255	4387	17207	0.955	775 sec
0.64	43	346	1779	6300	0.954	573 sec
0.66	41	222	776	2491	0.953	549 sec
0.68	36	127	293	832	0.947	446 sec
0.7	33	70	128	331	0.946	376 sec
0.72	25	39	52	118	0.938	258 sec
0.74	18	20	24	52	0.932	173 sec
0.76	12	9	9	18	0.928	90 sec
0.78	10	4	4	8	0.923	69 sec
0.8	5	1	1	2	0.921	36 sec

language and the experiments have been done on a laptop with a Core i5 CPU, 3MB cache, 4GB RAM, and a Windows 8 operating system. The obtained results demonstrate that the execution time of the proposed algorithm is reasonable and it is highly affected by the *MinSup* value.

4.3 Comparisons With Other Studies

Since there are fundamental differences between *SWApriori* and the previous studies -in terms of the existence of transactions, data conversion, and structure of the items- in this section *SWApriori* and its generated results are compared structurally with those of the studies in [20], [24], and [25].

The study in [24] uses only two parts of triples to generate transactions and ignores the remaining part. Thus, this approach loses a lot of information. In addition, since in this approach, one part of the triples is used as TID and this TID is employed by the traditional Apriori algorithm just for identifying the items of transactions, the generated rules are ambiguous since no information about TIDs is presented in the generated rules. *SWApriori* generates all ARs that can be generated by [24] when objects are used as *Target* (the 2nd and 4th rows of Table 1). Similarly, the work presented by [25] assumes a tabular structure for the input semantic web dataset, and by ignoring the relations between items, generates ARs using only subjects (as TID) and objects (as transaction items).

In the algorithm proposed in [20], which we call it *MPAR*, first, the type of items that are used to construct transactions are determined (by using *rdf:type*). Then, the objects that are of that type and are reachable by a common subject are employed to generate transactions. Thus, the items of the transactions consist of values of the objects. In this algorithm, the end-user should have deep knowledge about the structure of the dataset and its underlying ontology to determine the source subjects as well as the objects that are used to generate transactions. This approach is useful for cases where the items of the transactions can be interpreted unambiguously. For example, by reading the item "Methotrexate", a domain expert can understand that it is a drug as the prescription of a visit. However, for example, it is ambiguous to understand the real meaning of an item when just a country name (*e.g.*, Russia) appears in the item without its incoming relation (*e.g.*, *BornIn:Russia* vs. *NeighborEast:Russia*).

In this section, another experiment has been conducted on *DBPedia* to demonstrate that in contrast to the MPAR algorithm, the presented SWApriori algorithm can generate meaningful ARs from general datasets without the end-user intervention. In this experiment, the entities whose type is Country ($rdf : type\ dbo : Country$) have been considered by $MPAR$ to generate the items of transactions. In this regard, a SPARQL command is executed over *DBPedia*[5] to extract such entities that are reachable from other countries as subjects. In addition, the predicates that correspond to the subjects to the entities are extracted

Table 5. Some Large Itemsets Generated by SWApriori and MPAR [20].

SWApriori	**MPAR**
era(New_Imperialism), governmentType(Constitutional_monarchy)	New_Imperialism, Constitutional_monarchy
ethnicGroup(Multiracial), status(British_Overseas_Territories)	Multiracial, British_Overseas_Territories
empire(Holy_Roman_Empire), era(Renaissance)	Holy_Roman_Empire, Renaissance
east(Russia), establishedEvent(European_Union)	Russia, European_Union
east(Russia), northEast(Russia)	Russia, Russia
establishedEvent(European_Union), southEast(Belarus)	European_Union, Belarus
empire(Former_Qin), empire(Later_Zhao)	Former_Qin, Later_Zhao
empire(Later_Qin), empire(Northern_Wei)	Later_Qin, Northern_Wei
cultures(Kish_civilization), epochs(Bronze_Age), location(Syria)	Kish_civilization, Bronze_Age, Syria

as well to feed to the *SWApriori* algorithm. Then, the *SWApriori* and *MPAR* algorithms have been applied to the obtained triples and the generated transactions, respectively. Since the transactions generated for the *MPAR* algorithm contain only the values of entities without their incoming relation, with a given *MinSup* value, the *MPAR* algorithm generates more itemsets with a higher support value (w.r.t. the itemsets generated by *SWApriori*). However, the itemsets generated by *SWApriori* are more specific and meaningful. Table 5 shows some large itemsets generated by both *SWApriori* and *MPAR* algorithms. As this table shows, since the *MPAR* algorithm uses just the values of entities to generate ARs, some large itemsets obtained from general datasets are ambiguous. Moreover, in cases where a given object has different types of incoming relations, it is very hard to interpret the obtained frequent itemsets even by a domain expert (*e.g.*, see the 5^{th} row in Table 5 which denotes that many countries whose eastern country is Russia, their northeastern country is Russia as well).

5 CONCLUSIONS AND FUTURE WORK

In this paper, a new algorithm named *SWApriori* has been proposed to mine ARs from semantic web data. The ARs are discovered directly without the end-user intervention and without converting the semantic web data to transactional-based traditional ones. The proposed algorithm deals with all kinds of datasets and ontologies presented in the triple format regardless of the dataset domain.

The proposed solution first extracts large objects from the input dataset and then generates 2-large itemsets from large objects regardless of the concept of transactions. Each generated itemset consists of multiple items and each item consists of one entity and one relation. Afterward, each set of L-large itemsets ($L \geq 3$) is generated from (L-1)-large itemsets. Finally, the ARs are generated from the large itemsets.

The efficiency of the proposed solution has been demonstrated by conducting several experiments on semantic web datasets *Factbook* and *DBPedia*. The obtained results show that the proposed solution could generate relational ARs from semantic web data directly by considering the inherent complexity of these data. Supplementary experiments have shown that the proposed technique can generate interesting ARs that are not discoverable by state-of-the-art ARM techniques. The positive features of the proposed algorithm are as follows:

- It does not need transactions. Thus, it can be applied to any dataset containing triples.
- It considers different relations between entities: each generated item consists of one entity and one relation. These items can then be used to generate ARs.
- It does not convert semantic web data to traditional ones: The input dataset can be used in its original format (triple format).
- It handles the heterogeneous structure of the semantic web data: a linked list-based data structure has been used to store and organize the input data.
- It does not involve the end-user in the mining process: in the presented algorithm, the main role of the end-user is to provide the input dataset and the values of *MinSup* and *MinConf*. In other words, the end-user does not need to be a domain expert and be aware of the dataset and ontology structure. However, if the end-user wishes, he/she can filter the input dataset by SPARQL commands.

We believe that this kind of unsupervised learning will become important in the future and will affect the machine learning research area, especially the area of semantic web research. As future work, we intend to extend the proposed technique to apply it to linked data. It will have different challenges, such as ontology alignment, ontology mapping, and broken links.

References

[1] C. Bobed, P. Maillot, P. Cellier, and S. Ferré. Data-driven assessment of structural evolution of RDF graphs. *Semantic Web*, page 1–23, April 2020. ISSN 2210-4968. doi:10.3233/SW-200368. URL `http://doi.org/10.3233/SW-200368`.

[2] P. Ristoski. *Exploiting semantic web knowledge graphs in data mining.* IOS Press, 2019.

[3] X. Liu, K. Zhai, and W. Pedrycz. An improved association rules mining method. *Expert Systems with Applications*, 39(1):1362–1374, 2012. doi:10.1016/j.eswa.2011.08.018. URL `https://doi.org/10.1016/j.eswa.2011.08.018`.

[4] K. Yan, W. Cui, and T. Zhao. Frequent Pattern-based Graph Exploration. In *Proceedings of the 12th International Symposium on Visual Information Communication and Interaction.* ACM Press, 2019. doi:10.1145/3356422.3356446. URL `https://doi.org/10.1145/3356422.3356446`.

[5] C.S.R. Prabhu, A. S. Chivukula, A. Mogadala, R. Ghosh, and L.M. Jenila Livingston. Social Semantic Web Mining and Big Data Analytics. In *Big Data Analytics: Systems, Algorithms, Applications*, pages 217–231. Springer Singapore, 2019. doi:10.1007/978-981-15-0094-7_7. URL `https://doi.org/10.1007/978-981-15-0094-7_7`.

[6] Chengqi Zhang and Shichao Zhang. *Association rule mining: models and algorithms.* Springer, 2003.

[7] T. Herawan and M. M. Deris. A soft set approach for association rules mining. *Knowledge-Based Systems*, 24(1):186–195, February 2011. doi:10.1016/j.knosys.2010.08.005. URL `https://doi.org/10.1016/j.knosys.2010.08.005`.

[8] G. Barisevičius, M. Coste, D. Geleta, D. Juric, M. Khodadadi, G. Stoilos, and I. Zaihrayeu. Supporting Digital Healthcare Services Using Semantic Web Technologies. In *Lecture Notes in Computer Science*, pages 291–306. Springer International Publishing, 2018. doi:10.1007/978-3-030-00668-6_18. URL `https://doi.org/10.1007/978-3-030-00668-6_18`.

[9] T. Osadchiy, I. Poliakov, P. Olivier, M. Rowland, and E. Foster. Recommender system based on pairwise association rules. *Expert Systems with Applications*, 115:535–542, January 2019. doi:10.1016/j.eswa.2018.07.077. URL `https://doi.org/10.1016/j.eswa.2018.07.077`.

[10] G. F. Pelap, C. F. Zucker, F. Gandon, and L. Polese. Web Semantic Technologies in Web Based Educational System Integration. In *Lecture Notes in Business Information Processing*, pages 170–194. Springer International Publishing, 2019. doi:10.1007/978-3-030-35330-8_9. URL `https://doi.org/10.1007/978-3-030-35330-8_9`.

[11] M. A. Valle, G. A. Ruz, and R. Morrás. Market basket analysis: Complementing association rules with minimum spanning trees. *Expert Systems with Applications*, 97:146–162, May 2018. doi:10.1016/j.eswa.2017.12.028. URL `https://doi.org/10.1016/j.eswa.2017.12.028`.

[12] H. Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3):489–508, December 2016. ISSN 2210-4968. doi:10.3233/SW-160218. URL `http://doi.org/10.3233/SW-160218`.

[13] S. Muggleton and L. d. Raedt. Inductive Logic Programming: Theory and methods. *The Journal of Logic Programming*, 19-20:629–679, 1994. doi:10.1016/0743-1066(94)90035-3. URL `https://doi.org/10.1016/0743-1066(94)90035-3`.

[14] L. A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web.* ACM Press, 2013. doi:10.1145/2488388.2488425. URL `https://doi.org/10.1145/2488388.2488425`.

[15] L. Galárraga, C. Teflioudi, K. Hose, and F. M. Suchanek. Fast rule mining in ontological knowledge bases with AMIE+. *The VLDB Journal*, 24(6):707–730, July 2015. doi:10.1007/s00778-015-0394-1. URL `https://doi.org/10.1007/s00778-015-0394-1`.

[16] B. T. Luong, S. Ruggieri, and F. Turini. Classification Rule Mining Supported by Ontology for Discrimination Discovery. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW).* IEEE, 2016. doi:10.1109/icdmw.2016.0128. URL `https://doi.org/10.1109/icdmw.2016.0128`.

[17] S. Vojíř, V. Zeman, J. Kuchař, and T. Kliegr. EasyMiner.eu: Web framework for interpretable machine learning based on rules and frequent itemsets. *Knowledge-Based Systems*, 150:111–115, June 2018. doi:10.1016/j.knosys.2018.03.006. URL `https://doi.org/10.1016/j.knosys.2018.03.006`.

[18] J. S. Hong. A Methodology for Searching Frequent Pattern Using Graph-Mining Technique. *Journal of Information Technology Applications and Management*, 26(1):65–75, 2019. ISSN 2508-1209.

[19] A. V. V. Rao and B. E. Rambabu. Association rule mining using FPTree as directed acyclic graph. In *IEEE-International Conference On Ad-*

vances In Engineering, Science And Management (ICAESM-2012), pages 202–207. IEEE, 2012.

[20] V. Nebot and R. Berlanga. Finding association rules in semantic web data. *Knowledge-Based Systems*, 25(1):51–62, February 2012. doi:10.1016/j.knosys.2011.05.009. URL `https://doi.org/10.1016/j.knosys.2011.05.009`.

[21] W.X. Wilcke, V. d. Boer, M.T.M. d. Kleijn, F.A.H. v. Harmelen, and H.J. Scholten. User-centric pattern mining on knowledge graphs: An archaeological case study. *Journal of Web Semantics*, 59:100486, 2019. doi:10.1016/j.websem.2018.12.004. URL `https://doi.org/10.1016/j.websem.2018.12.004`.

[22] A. S. Heydari Yazdi and M. Kahani. A novel model for mining association rules from semantic web data. In *2014 Iranian Conference on Intelligent Systems (ICIS)*. IEEE, February 2014. doi:10.1109/iraniancis.2014.6802574. URL `https://doi.org/10.1109/iraniancis.2014.6802574`.

[23] R. Ramezani, M. Saraee, and M. A. Nematbakhsh. MRAR: mining multi-relation association rules. *Journal of Computing and Security*, 1(2):133–158, 2014. ISSN 2322-4460.

[24] Z. Abedjan and F. Naumann. Improving RDF Data Through Association Rule Mining. *Datenbank-Spektrum*, 13(2):111–120, 2013. doi:10.1007/s13222-013-0126-x. URL `https://doi.org/10.1007/s13222-013-0126-x`.

[25] E. Bytyçi, L. Ahmedi, and F. A. Lisi. Enrichment of association rules through exploitation of ontology properties – healthcare case study. *Procedia Computer Science*, 113:360–367, 2017. doi:10.1016/j.procs.2017.08.345. URL `https://doi.org/10.1016/j.procs.2017.08.345`.

[26] V. Narasimha, P. Kappara, R. Ichise, and O. Vyas. Liddm: A data mining system for linked data. In *Workshop on Linked Data on the Web. CEUR Workshop Proceedings*, page 108, 2011.

[27] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI Global, 2011. doi:10.4018/978-1-60960-593-3.ch008.

[28] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, July 2009. URL `http://tomheath.com/papers/bizer-heath-berners-lee-ijswis-linked-data.pdf`.

[29] R. Ramezani, M. Saraee, and M. A. Nematbakhsh. Finding association rules in linked data, a centralization approach. In *2013 21st Iranian Conference on Electrical Engineering (ICEE)*. IEEE, May 2013. doi:10.1109/iraniancee.2013.6599550. URL `https://doi.org/10.1109/iraniancee.2013.6599550`.

[30] M. A. Khan, G. A. Grimnes, and A. Dengel. Two pre-processing operators for improved learning from semanticweb data. In *First RapidMiner Community Meeting And Conference (RCOMM 2010)*, 2010.

[31] C. Kiefer, A. Bernstein, and A. Locher. Adding Data Mining Support to SPARQL Via Statistical Relational Learning Methods. In *Lecture Notes in Computer Science*, pages 478–492. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-68234-9_36. URL `https://doi.org/10.1007/978-3-540-68234-9_36`.

[32] P. S.M Tsai and C. Chen. Mining interesting association rules from customer databases and transaction databases. *Information Systems*, 29 (8):685–696, December 2004. doi:10.1016/s0306-4379(03)00061-9. URL `https://doi.org/10.1016/s0306-4379(03)00061-9`.

[33] A. Patel and S. Jain. Present and future of semantic web technologies: a research statement. *International Journal of Computers and Applications*, pages 1–10, January 2019. doi:10.1080/1206212x.2019.1570666. URL `https://doi.org/10.1080/1206212x.2019.1570666`.

[34] M. Barati, Q. Bai, and Q. Liu. Mining semantic association rules from RDF data. *Knowledge-Based Systems*, 133:183–196, 2017. doi:10.1016/j.knosys.2017.07.009. URL `https://doi.org/10.1016/j.knosys.2017.07.009`.

Reza Ramezani has received the B.Sc. and M.Sc. degrees in computer engineering from Shiraz Faculty of Engineering and Isfahan University of Technology, Iran, in 2010 and 2012, respectively. Since 2014, he collaborates with the Departamento de Computadores y Automatica (DACyA), at the Universidad Complutense de Madrid (UCM), Madrid, Spain. He received the degree of Ph.D. of Software Engineering from Department of Computer Engineering, Ferdowsi University of Mashhad (FUM), Mashhad, Iran in Jan 2017. Reza is currently an Assistant Professor with the Department of Software Engineering, University of Isfahan (UI), Isfahan, Iran. His main research area includes data mining, semantic web, and text analysis.

Mohammad Ali Nematbakhsh is a full professor in software engineering department at University of Isfahan. He received his BSc in electrical engineering from Louisiana Tech University, USA, in 1981, and his MSc and PhD degrees in electrical and computer engineering from the University of Arizona, USA, in 1983 and 1987, respectively. He worked for Micro Advanced Co. and Toshiba Corporation for many years before joining University of Isfahan. He has published more than 150 peer reviewed research papers, several U.S. registered patents and two database books that are widely used in universities. His main research interests include intelligent Web and bigdata technology.

Mohamad Saraee holds a chair in Data Science at the School of Science, Engineering and Environment, University of Salford-Manchester. He received his PhD in Computer Science from University of Manchester and MSc in Computer Engineering from University of Wyoming, USA. He has 25 years+ experience in the areas of Data Science, Big Data Analytics and Data/Text Mining. Professor Saraee has an established track-record in applying AI and data mining in the medical and financial domain and Self-Driving Cars. He has been involved in a number of Data Science funded projects totalling £700K+ as principals and/or co-investigator.