A Survey on Trust and Reputation Models for Web Services: Single, Composite, and Communities

Omar Abdel Wahab[a], Jamal Bentahar[a☆], Hadi Otrok[ba], and Azzam Mourad[c]
[a]Concordia University, Montreal, Canada
[b]Khalifa University of Science, Technology & Research, Abu Dhabi, UAE
[c]Lebanese American University, Beirut, Lebanon

Abstract

Web services selection constitutes nowadays a major challenge that is still attracting the research community to work on and investigate. The problem arises since decision makers (1) cannot blindly trust the service or its provider, and (2) ignore the environment within which the service is operating. The fact that no security mechanism is applicable in such a completely open environment, where identities can be easily generated and discarded makes social approaches such as trust and reputation models appealing to apply in the world of Web services. This survey classifies and compares the main findings that contributed in solving problems related to trust and reputation in the context of Web services. First, a high-level classification scheme partitions Web services into three main architectures: single, composite, and communities. Thereafter, a low-level classification within each architecture categorizes the trust and reputation models according to the technique used to build the trust value. Based on this classification, a profound analysis describing the advantages and shortcomings of each class of models is presented; leading to uncover possible topics that need further study and investigation. In particular, we discuss the challenging problem of having active malicious Web services in the composite and community-based architectures. Thus, the paper can be used by the future researchers as a roadmap to explore new trust and reputation models for Web services taking into account the shortcomings of the existing models.

Keywords: Web service, architecture, trust, reputation, security, decision making.

1. Introduction

1.1. Background

Web services are gaining nowadays an increasing attention due to their ability to achieve efficient and loosely-coupled cross-organizational business-to-business integration. According to the World Wide Web

Consortium W3C, a Web service is a *"software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols"*. The concept of Web services is based on a recipe of three ingredients that helps achieve business-to-business integration, which are the: (1) service-oriented architecture (SOA), (2) redesign of middleware protocols, and (3) standardization [1].

Web services are commonly represented and described over three main architectures [1, 2, 3]: single, composite, and communities. Single Web services are referred to as those services working individually to satisfy the users' requests. Composition involves assembling a set of individual services to achieve a complex functional and/or non-functional requirement that cannot be fulfilled by a single service [2]. Grouping the services offering the same functionality into communities [3] inherits the concept of clustering that has shown to be efficient in many domains [4, 5, 6]. The idea of such an architecture is to facilitate the discovery of Web services by enhancing their visibility and to increase their overall performance by allowing them to cooperate inside communities.

1.2. Problem Definition

Despite the bright future of Web services, this emerging technology encounters some challenges that constitute nowadays hot topics for the research community. Among them, Web services selection has attracted recently several contributions aiming at making optimal selections in this context. The problem arises when the user has to select, among a group of services offering the same functionality, the one that can best satisfy his requests. The usual security mechanisms such as authentication and access control cannot handle the problem of selection. These mechanisms stop at the borders of verifying credentials and checking identities but cannot foretell how well services will behave and perform. Recently, focus is heading towards the social approaches that are based on trust and reputation to replace the usual security mechanisms.

According to the Concise Oxford Dictionary, *"Reputation is what is generally said or believed about a person's or thing's character or standing"* [7]. Informally, reputation represents the combined measure of reliability inferred by feedback or ratings gathered from members in a certain community. Trust can be defined as *"a subjective probability an agent has about another's future behavior"* [8]. That is, the degree of trustworthiness one agent assigns to another agent/group of agents in performing a certain action [7]. The main difference between these two concepts is that trust is mainly a personal and subjective notion in contrary to the reputation that is public and combined. In other words, one agent A may still trust another

agent B despite B's bad reputation if A has a close relation to B or even has some private information about B that surpass B's public reputation. Numerous trust and reputation models have been proposed for many open systems [9, 10, 11, 12]. In the context of Web services, a trust and reputation model is a method that enables decision makers to distinguish good services from bad ones based on users' feedbacks. The importance of trust and reputation models in the context of Web services stems from their ability to enable users and service providers to differentiate among the services that offer similar functionalities on the basis of how well these services behaved in the past history. This helps them make thoughtful selections and avoid the bad choices since making random choices in such an open system may expose users/providers to quality, cost, and even security problems. Practically, any provider has the freedom to publish bad-quality, expensive, and even harmful services; which makes wise selections of great importance.

1.3. Contributions

Several reviews [13, 14, 15, 16] targeting trust and reputation in Web services have been advanced. The motivation for this survey stems from two main reasons. First, the existing survey papers lack for a comprehensive view of Web services' architectures. To the best of our knowledge, this is the first review paper that classifies Web services according to their architecture and provides a collection of criteria that are important for the success of the trust and reputation models in each architecture. The second motivation is the lack of a profound and systematic review of trust and reputation in the domain of Web services. This paper presents a high-level classification scheme for the Web services according to their architectures and a sub-classification in each architecture based on the underlying technique used to construct the trust value. Moreover, we define for each architecture a set of criteria that are necessary for the success and effectiveness of the trust and reputation models targeting this architecture. The classification scheme aims to help (1) providers improve the quality and performance of their services, (2) customers enhance the quality and credibility of their ratings, and (3) research community study and investigate some open challenges that are not solved yet in this domain.

1.4. Research Methodology and Organization

The proposed criteria are selected to answer a collection of research questions we raise for each architecture of Web services. These questions target the major challenges that each architecture may face in the context of trust and reputation. The challenges have been identified as those ones that received most of the attention in the papers used for the survey. Many of these challenges are also explicitly identified as major

issues in the surveyed papers. In the single architecture, most of the research is oriented to tackle the issues of bootstrapping, credibility of ratings, trust dynamism, and representativeness of the trust and reputation sources [17, 18, 19, 20, 21, 22]. For the composite architecture, the identified major challenges are determining the contribution of each component in the composition process and the problem of task allocation among components [23, 24, 25, 26, 27]. In the community-based architecture, joining communities and the influence of that joining on the performance and reputation of the community have been the key challenges [28, 29, 30, 31, 32]. Moreover, we have noticed that the topic of malicious attackers, that have major impacts on the trust and reputation of Web services and that has been a major challenge in many important domains such as networks [33, 34], has been disregarded in the context of Web services. To this end, we raise this topic and highlight its importance by means of profound analysis and simulation experiments. In summary, the main challenges in the single architecture are related to the quality and credibility of the procedure used to build the trust/reputation values. For the composite architecture, the challenges are expanded to cover the issues of estimating the contribution and performance of the Web service constituents in the composition process as well as the task allocation problem among these constituents and the security concerns that may be engendered by the malicious constituents. As for the community-based architecture, the issues of making thoughtful joining strategies for the communities and protecting them against malicious attacks are additional concerns. Numerous criteria exist in other surveys, where each survey focuses on certain aspect(s) related to trust and reputation. Similar to these surveys, we do not claim to cover all the criteria needed for the trust/reputation models; but our criteria are defined to answer the proposed research questions. It is worth mentioning as well that numerous criteria proposed in other surveys, even not explicitly expressed in our work, can be inferred by combining some of our proposed criteria. Other criteria such as those related to the efficiency and complexity of the reputation systems and aggregation algorithms are out of the scope of this study as the approaches selected for comparison do not consider these aspects.

The approaches chosen for comparison in each Web services architecture are selected from papers published between 2009 to 2014 in refereed journals and international conferences. Moreover, we included the papers that are major (based on the number of citations) in the domain of trust and reputation in Web services published before 2009 and that are important to understand the core of the topic such as [17, 21]. The objective is not to gather and compare all the approaches that tackled trust and reputation in Web services. The reason is that we advance a two-phase classification of the current approaches and compare each class of models based on the proposed criteria, where approaches in the same class share the same basic idea but

differ in some minor and technical details.

The structure of the paper is organized as follows. Section 2 explains the need for trust and reputation models in the field of Web services and raises the research questions that our study aims to address. Section 3 compares our survey against the other survey papers in the same domain and highlights the unique features of our work. Section 4 classifies and compares the main trust and reputation models proposed for Web services. Section 5 discusses the limitations of the existing approaches and suggests possible research topics in this regard. Finally, Section 6 concludes the paper.

2. Problem Statement: A Real-life Scenario

In this section, we illustrate the need for a trust and reputation model in the scope of Web services by describing a real-life scenario and raise the research questions that our work aims to address. Consider the case of flight booking application. A customer makes a request containing the flight dates, origin and destination, type of tickets (one way or return), and number of guests to the *Flight Booking* Web service and asks for the information related to such a flight (i.e., companies, timing, prices). To gather such information, the *Flight Booking* Web service has to make a series of invocations. Practically, it would inquire the name of the companies, ticket prices, and timing on the specified route from the *Airline Reservation* service. Moreover, it will contact the *Hotel Reservation* service to get the prices and availabilities of hotel accommodations in the given destination. It will also invoke the *Car Rental* to get the options and prices of cars.

In this scenario, two types of interactions take place: (1) customer to service explicit interaction, and (2) service to service transparent interaction. The first type of interactions refers to the single architecture of Web services, while the second describes the composite architecture. Although the second type of interactions does not impact the customer directly, it will affect the quality of the whole transaction. In fact, the overall quality of a composite Web service is affected by the quality perceived by each single service in this composition. Suppose that the *Hotel Reservation* Web service is overloaded by a huge number of requests. This would increase the response time of the overall transaction. Therefore, as much as the customer is interested in the selection of the appropriate service to obtain the "best" possible quality, the *Flight Booking* service is interested in selecting the appropriate Web services to be part of the composition in a way that allows him to maintain a good record among other *Flight Booking* Web services. Thus, both the customer and *Flight Booking* Web service have to make appropriate decisions in this context. Such a decision cannot be made randomly due to the fact that in such an open environment, anyone can offer services that may be

of low quality, time consuming, expensive or even harmful. This raises the need for mechanisms that enable decision makers to distinguish well from bad services. Applying the usual security mechanisms such as authentication and access control cannot help us make optimal decision in such a case. In fact, learning the credentials of Web services is not enough to predict how well these Web services will perform, but having an idea about their past interactions would signal their trustworthiness. Without a reputation-based selection, it would be difficult for both customer and provider to select the appropriate services to deal with.

By using a reputation-based mechanism, the customer is increasing his chance to get higher Quality of Service (QoS) referred to as the overall performance perceived from Web services. The provider, in his turn, is decreasing the risk of getting distorted because of non-reputable external components. While achieving these goals, several challenges arise. Some of these challenges are generic for all the Web services, while others are specific for each architecture. For the single Web services' architecture, the main challenges can be summarized by the following research questions:

1. **RQ1**: How and based on which parameters to evaluate the reputation of the Web services?
2. **RQ2**: How to assign initial trust values for the new Web services?
3. **RQ3**: How to adapt the trust values to the dynamic change in the Web services' performance?
4. **RQ4**: How to protect the trust/reputation values against collusion and deception problems?

These challenges apply as well in the composite architecture in addition to supplementary challenges imposed by the composite architecture such as:

5. **RQ5**: How to evaluate the performance of the individual constituents in the overall composite service?
6. **RQ6**: How to assess the trust of the constituents when their performance cannot be fully observable?
7. **RQ7**: How to manage the collaboration and task allocation issues among the constituents?
8. **RQ8**: How do malicious constituents affect the reputation and performance of the composite service?

In the community-based architecture, several Web services offering the same functionality are grouped into clusters to ease their discovery process and increase the overall performance. Thus, if a *Hotel Reservation* Web service, say H_1, does not have enough QoS metrics such as throughput or response time to fulfill the request coming from the *Flight Booking* Web service, it can cooperate with the other community members offering the same functionality (e.g., H_2 and H_3) or delegate the request for them to perform it with better performance. In such an architecture, dealing with trust and reputation becomes more ramified and more issues to consider, in addition to those of the single and composite architectures, arise such as:

9. **RQ9**: How to evaluate the reputation of a community in such a dynamic environment where Web service continuously join and leave?
10. **RQ10**: Would the community members cooperate with each other and why? How does this affect their reputations and the reputation of the whole community?
11. **RQ11**: How and based on what to select the Web services to be part of the community?
12. **RQ12**: How do malicious Web services influence the reputation and performance of the community?

Several approaches were proposed trying to answer some of these questions, while other questions still need further study and investigation. In what follows, we present and classify the main contributions that addressed issues related to trust and reputation in these three architectures, derive a collection of criteria for the trust and reputation models in each architecture from the aforementioned questions, compare the class models, and identify a gap from which researchers can find important topics to work on and explore.

3. Related Work

Several surveys can be found in the literature about trust and reputation in Web services [13, 14, 15, 16]. Wang et al. proposed in [13] a classification scheme for trust and reputation systems in Web services based on three criteria: (1) centralized or decentralized, i.e., there exists a central party charged of managing the reputation for all the members or not; (2) person or resource, i.e., they target persons or resources; and (3) global or personalized, i.e., collected based on opinions from general population that is visible to all members or based on opinions from group of members.

In [14], the authors focused on the trust management models and issues related to semantic Web services. They classified the trust models based on the way used to compute the trust value; resulting in three categories: (1) Trust Computation Related to Services, where services establish trust for each other; (2) Trust Computation on Consumer View, where consumers provide feedback on the services based on their interactions; and (3) Trust Computation for Content and Context, which uses meta-data information to analyze the semantic data published on the Web.

In [15], the authors present a comparison summary between the reputation-based approaches proposed in the Service-Oriented Computing domain based on four criteria: maturity, majority, cost, and infrastructure. The maturity stresses the need for users' ratings when building trust. Majority points out that a certain trust mechanism should be independent from the credibility of the majority of ratings that may be dishonest. Cost refers to the complexity and extensibility of the trust mechanism, while infrastructure refers to the ability to

Table 1: Criteria for the trust and reputation models in the single Web services' architecture

ID	Criterion	References
C1	Cover multiple Quality of Service (QoS) metrics such as response time, throughput, and availability.	[17, 20]
C2	Consider the user preferences.	[15, 20]
C3	Account for both subjective and objective perspectives.	[20, 35, 36]
C4	Assess the credibility of raters.	[20, 35, 36]
C5	Have a bootstrapping mechanism.	[37, 27]
C6	Consider the trust dynamism.	[17, 27]
C7	Be independent from the credibility of the majority of ratings.	[15, 35, 36]

support distributed infrastructure such as Web services. In our work, maturity and majority are expressed in criteria $C2$ and $C7$ (Table 1) respectively.

Dragoni proposed in [16] a classification scheme for the trust-based Web services selection approaches based on their rationale; resulting in three classes: (1) Direct experience-based approaches in which consumers use the direct past experience with a certain service to build the trust for that service; (2) Trusted Third-Party (TTP) approaches in which consumers consult a trusted third party to build a trust for a certain service; and (3) Hybrid approaches that combine techniques from the two aforementioned classes to build integrated frameworks.

More generally than Web services, the topic of trust and reputation in online systems has been tackled in many review papers [7, 35, 36]. In [7], the authors presented a broad discussion about the notions of trust and reputation and proposed a classification for the trust and reputation models based on the reputation computation engines; resulting in six classes: Simple Summation or Average of Ratings, Bayesian Systems, Discrete Trust Models, Belief Models, Fuzzy Models, Flow Models. The focus of this survey is to discuss the trust and reputation in the deployed systems such as security and commerce rather than systematically reviewing the research literature.

In [35], the authors proposed a reference model for building reputation systems for e-services. They introduced a collection of criteria whose main objective is to ensure that the assessed reputation values reflect the actual trustworthiness of users. Several criteria may be inferred by combining criteria $C3$, $C4$, and $C7$ (Table 1) in our work. Examples of these criteria include: reputation should be assessed using a sufficient amount of information, and reputation system should be able to discriminate incorrect ratings.

In [36], the authors target the centralized online reputation systems by proposing a structure and providing a set of criteria for each component in this structure. Some of these criteria, related to the quality of the ratings, are reflected in $C3$, $C4$, and $C7$ (Table 1) in our work. Other criteria focus on the efficiency of the reputation systems as well as the aggregation algorithms by highlighting some relevant requirements such

as complexity and robustness.

Similar to the aforementioned surveys, we do not claim to cover all the criteria needed for the trust and reputation models; however, our criteria are proposed to answer the research questions raised in Section 2. Numerous criteria proposed in other surveys, even not explicitly expressed in our work, can be inferred by combining some of our proposed criteria. Other criteria such as those related to the efficiency and complexity of the reputation systems and aggregation algorithms are out of the scope of this study as the approaches selected for comparison do not consider these aspects. Overall, the unique features of our survey are (1) defining Web services' architectures and describing their points of convergence and difference; (2) providing a sub-classification within each architecture on the basis of the technique used to build the trust/reputation value; (3) proposing a taxonomy of criteria for each architecture and comparing the class models and approaches in each architecture based on these criteria; and finally (4) discussing the limitations and future directions specific to each of these architectures.

4. Trust and Reputation in Web Services

Several trust and reputation models were proposed for Web services tackling a variety of topics. In this section, we present a high-level classification for the trust and reputation models according to the architecture of Web services they target and a low-level classification in each architecture based on the technique used to build the trust value. The classification scheme is depicted in Figure 1. We define as well a set of criteria for trust and reputation models in each architecture; based on which we conduct a high-level comparison among the classes of models and a low-level comparison among the major approaches in each architecture.

4.1. Single Web Services

Single Web services are referred to as those Web services working in a standalone manner to achieve users' requests. Trust and reputation models proposed for this architecture aim mainly to help users select the appropriate Web service that best achieves their requests. The following criteria are important for the trust and reputation models while achieving this goal [20, 37, 27, 35, 15, 17, 36]:

- **Criterion** #1: Cover multiple QoS metrics (e.g., response time, throughput, availability, etc.) to enable users to well-distinguish among functionally-similar services.
- **Criterion** #2: Consider the user preferences since users may be interested in different quality metrics (i.e., one user may be interested in the response time while another user may look for lower cost).

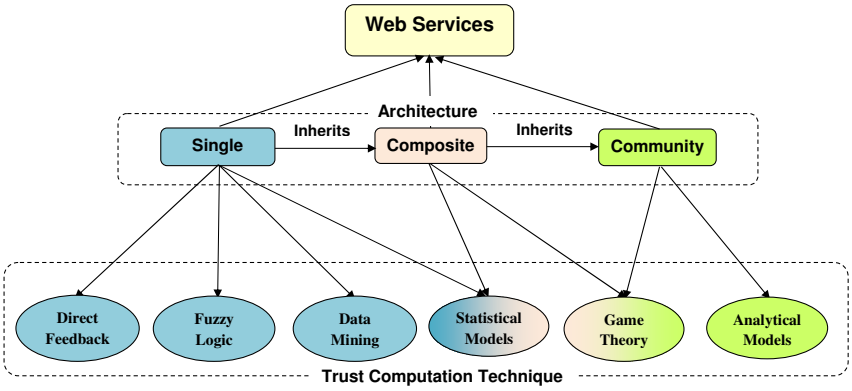


Figure 1: Classification scheme: Trust and reputation models are classified based on the architecture of Web services they target (high-level classification), and the technique they use to build the trust within each architecture (low-level classification)

- **Criterion** #3: Account for both subjective (feedbacks from users) and objective (QoS monitoring) perspectives while evaluating the trust and reputation of Web services.
- **Criterion** #4: Assess the credibility of raters to avoid collusion and deception.
- **Criterion** #5: Have a bootstrapping mechanism to assign initial trust values for the newcomer Web services (i.e., newly deployed Web services).
- **Criterion** #6: Consider the trust dynamism issue since the performance of Web services is subject to change over the time (ameliorate or deteriorate).
- **Criterion** #7: Avoid the dependency between the recommendation given to a certain service and the credibility of the majority of ratings.

These criteria are summarized in Table 1. Most of the trust and reputation models proposed for the single architecture of Web services use direct feedback collected from users to compute the trust value for the Web services. Few statistics-based, fuzzy-logic-based, and data-mining-based models were proposed for this purpose. More details on these models and their associated approaches are given in what follows. Thereafter, Table 2 compares the discussed approaches according to the criteria presented in Table 1.

4.1.1. Feedback-based models

Feedback-based models [17, 18, 38, 39] rely on the idea of collecting reviews concerning a certain Web service. These reviews are used then to build a trust value for the Web service in question. The source of reviews is either the provider or the consumer [40]. Provider-generated information include the descriptions

of the service recorded in the service registry. Consumer-generated information are, on the other hand, online reviews provided by the users who had dealt with the service during past interactions. For example, Maximilien and Singh [17] proposed a multi-agent framework based on ontology allowing providers to proclaim their services, users to state their preferences, and ratings about services to be built and shared. The ratings are based on the QoS metrics, which include well-known computing parameters such as latency and throughout but may involve also application-specific parameters such as shipping delay. The proposed framework relies on three main concepts: provider quality advertisement, customer quality preference, and service reputation. Using the provider quality advertisement, providers advertise their services by specifying the minimum and maximum possible quality values for the offered service as well as the promised value for this service. Consumers, in their turn, describe their preferences by specifying the minimum and maximum acceptable quality thresholds as well as the preferred quality value. Thereafter, a trust function is formulated based on the reputation function, the consumer's preferences, and the provider's advertisements. The aim of this function is to rank the services based on how well they satisfy users' requirements in order to help make selections. The framework also provides a mechanism to periodically monitor the services in order to allow users to replace the poorly-performing services by other well-performing ones.

Although feedback-based models have the advantage of considering the opinions of the users, which tends to be the most rational and meaningful metric for building the reputation of any service, these models suffer from major problems. First, feedback-based models provide no bootsrapping mechanism for computing initial trust for Web services. Second, the quality and credibility of the ratings is a main problem that encounters feedback-based models. More precisely, providers tend to hide the bad characteristics of their services and stress the good aspects for marketing and commercial purposes. On the other hand, the feedback provided by the consumers tend to be more realistic due to two main reasons. Firstly, the feedback presented by the consumers are usually user-oriented in the sense that they focus on the aspects that concern the user such as QoS and cost in contrast to the providers that tend to proclaim the service-oriented information. The second reason is that consumers have higher probability than providers of mentioning the weaknesses along with the strengths of the services as they are assumed to be neutral parties who have no direct interest in the promotion/demotion of certain services. However, this does not mean that the reviews presented by the consumers are always truthful. In fact, consumer-based reviews are usually not organized in a standard manner in the sense that each user has his own style in writing the reviews that is different from other users (e.g.,$\{0, 1, 2\}$ vs. $\{$excellent, good, bad$\}$). Besides, users usually tend to refrain from submitting

reviews as they have no incentives for doing so, which leads to biased computation of the aggregated trust value. Most importantly, consumers are rational agents who may be tempted to provide dishonest feedback resulting in benefit for them as a result of a certain collusion scenario. For example, some consumers may collude with the providers to submit positive feedback on their services and/or negative feedback on the services of their competitors versus obtaining reduced service fees.

This problem was tackled by several approaches [38, 18, 19], where the authors consider the existence of malicious raters that may provide untrustworthy ratings. The main limitation of these approaches is that they are based on the idea that the majority of raters are credible in the sense that the rating of a certain consumer is assumed trusted if it agrees with the majority of ratings and untrusted otherwise. In this way, malicious raters can still impose their opinions and get high reputations by merely submitting a large number of fake feedback in way that allows them to form the majority.

4.1.2. Statistics-based models

In general, statistical models [41] are used to describe the relationship among a set of variables by means of mathematical equations. In the context of single Web services, few statistical models [20, 42, 19] are used to compute trust values for the Web services. These models attempt to overcome the problems of the feedback-based models, which rely solely on the reviews provided by providers and/or users and that may be incredible, by considering multiple sources of trust and using statistical methods to combine them.

For example, Nguyen et al. [20] proposed a trust model based on Bayesian Network (BN) that integrates both subjective and objective trust sources such as: direct opinion (ratings from users), recommendation (combination of public and personalized metrics), and conformance (between promised and actual QoS values). Based on these sources, the final trust value is calculated as the weighted sum of the three metrics.

In RATEWeb [19], the authors proposed a set of metrics inspired by the social networks methodologies with the aim of enhancing the accuracy and dynamically assessing the changing conditions. These metrics involve the credibility of the raters (to target malicious ratings), personalized preferences (weighted preferences over the QoS metrics), temporal sensitivity (to assign more weight to the most recent ratings), and first-hand knowledge (to cope with Web services' performance dynamism). Finally, a statistical technique is used to combine these metrics and compute the trust value.

Although statistics-based models provide powerful mechanisms for building the trust value by collecting and combining multiple sources of trust, these models still cannot compute initial trust values for the

newcomer Web services as they provide no bootstrapping mechanism that tackles this problem.

4.1.3. Fuzzy-logic-based models

Fuzzy logic [43] is a reasoning approach that supports approximate rather than exact values. For the Web services, fuzzy models [21, 44] are used to analyze the semantic and rationale behind the feedback left by the users. The motivations behind this are to (1) facilitate the construction of recommendations by aggregating the feedback left by users having the same preferences together, i.e., a user interested in the response time will be more interested in knowing the feedback related to the response time than those related to the price; (2) detect the bogus ratings provided by malicious users, e.g., users who are always submitting positive feedbacks on a certain Web service although the performance of this service was bad in multiple invocations. In [21], the authors proposed a fuzzy-logic-based reasoning model that combines both the subjective perspective represented by users' ratings and the objective perspective referred to as the compliance between promised and actual performance. To this end, they propose to assign a rating for each Web service based on its compliance value and compare it with users' ratings. This rating is computed in way that makes it biased towards a certain parameter (e.g., response time). Thereafter, the rating given by the user is compared against all the estimated ratings and the estimated rating that best matches the user's rating is deemed to be equivalent to the user's rating.

Although fuzzy-logic-based models try to understand the semantic behind the ratings provided by the users, which constitutes an important topic in the context of trust and reputation, these models offer only a set of rules and comparisons as ultimate output but provide no mechanism for computing the final trust value and are not able hence to help users and/or services make selections. They cannot compute initial trust values for the new services as well. Moreover, they do not take into account the dynamism of the trust.

4.1.4. Data-mining-based models

Data mining is an interdisciplinary subject that describes the process of extracting hidden patterns from huge datasets [45]. Data mining is becoming increasingly adopted in many domains such as medicine, engineering, science, business, etc. Despite its importance, this emergent discipline has not been well-exploited to address the problems related to trust and reputation in Web services. A data-mining-based approach was presented in [22], which uses the text mining to analyze the reviews provided by the users in order to evaluate the Web services and facilitate thus their selection. However, this approach is based on the naive assumption that the reviews presented by the users are always credible. Moreover, the authors didn't provide an in-depth

Table 2: Comparison summary between the main trust and reputation approaches in the single architecture

Approach	Model	C1	C2	C3	C4	C5	C6	C7
Maximilien and Singh [17]	Feedback-based	✓	✓	✓			✓	✓
Malik and Bouguettaya [18]	Feedback-based	✓	✓		✓		✓	
Malik and Bouguettaya [19]	Statistical	✓	✓		✓		✓	
Nguyen et al. [20]	Statistical	✓	✓	✓	✓			✓
Sherchan et al. [21]	Fuzzy logic	✓	✓	✓	✓			✓
Thurow et al. [22]	Data mining	✓	✓					✓

methodology of how the text mining will be effectively performed. Additionally, the bootstrapping and trust dynamism issues are ignored in this approach. Further steps are required leading to take advantage of the promising techniques offered by data mining (e.g., clustering, classification, frequent patterns, association rule, etc) [46], and that seem to be useful to solve problems related to trust and reputation.

4.2. Composite Web Services

Web services' composition involves integrating and organizing a set of services to achieve certain complex functional and/or non-functional requirements that cannot be accomplished by a single Web service [2]. In this architecture, trust and reputation models aim to help composition designers select the appropriate Web services to be part of the composition process resulting in benefit for both designers (better reputation) and users (better quality). To achieve this goal, several criteria have to be taken into consideration. As composite services are no more than a set of single Web services working together to achieve a certain objective, the requirements proposed for the single architecture (Table 1) apply as well for the composite architecture in addition to other important requirements such as [24, 23, 47, 48, 27, 26, 33, 34]:

- **Criterion** #8: Capture the responsibility of each constituent in the overall quality of the composite Web service in order to improve current compositions and facilitate future selections.
- **Criterion** #9: Consider that the responsibility of each constituent cannot be fully observed since users usually deal with the composite service as a monolithic entity.
- **Criterion** #10: Take into account the dynamism in the behavior of the constituents even when this change does not affect the overall quality of the composite service. For instance, consider the case of a service composed of two constituents X and Y. Initially, X is good and Y is bad. If X changes to bad and Y changes to good, the model should be able to capture this change although the overall performance of the composite service is not affected.
- **Criterion** #11: Monitor the variations in the QoS parameters of the constituents since predicting the performance based on the previous behavior cannot always yield reliable results as the performance

Table 3: Criteria for the trust and reputation models in the composite Web services' architecture

ID	Criterion	References
C8	Capture the responsibility of each constituent.	[23, 24, 48]
C9	Consider that the responsibility of each constituent cannot be fully observed.	[23, 24, 48]
C10	Take into account the dynamism of the behavior of the constituents even when this change does not affect the overall quality of the composite Web service.	[24]
C11	Monitor the variations in the QoS parameters of the constituent.	[47]
C12	Study the collaboration and task allocation among composite service's constituents.	[26, 27]
C13	Account for the active malicious constituents.	[33, 34]

can change in irregular manner (e.g., on demand).

- **Criterion** #12: Study the collaboration and task allocation problems among the constituents of the composite Web service to guarantee building reliable and well-performing compositions.

- **Criterion** #13: Consider the existence of *active malicious constituents* whose objective is to join some compositions and lunch attacks against the composite service or some partner constituents.

These criteria are summarized in Table 3. Numerous trust and reputation models have been advanced for the composite architecture. The dominant trend of these models use statistical techniques to compute the trust for the constituents, while the others employ game theory to model the collaboration and task allocation issues. More details about these models and their associated approaches are discussed in what follows. Thereafter, the discussed approaches are compared in Table 4 w.r.t the criteria presented in Table 3.

4.2.1. Statistics-based models

In the context of composite Web services, statistical models [25, 23, 49, 24, 50] have been widely used to model the relationships among the individual constituents and learn the responsibility of each constituent in the overall composite service. The objective is to help providers improve the quality of their existent compositions and make future selections. The challenges that led to the adoption of statistical models are the dynamic nature of the composite architecture and the difficulty of observing each constituent's quality. In fact, the dynamic aspect of the composition process makes it difficult to learn the order of the constituents. Moreover, the quality of each constituent cannot be always observed. For instance, when dealing with a hotel reservation service, the user may observe sometimes that a certain constituent always responds before the others. However, such information may not be always observable. Thus, statistical techniques are used to predict the quality of the constituents from the overall composite service's quality.

In [23], the authors employed Bayesian Network to assess the trustworthiness of the constituents through a reputation-based trust mechanism. Thus, a probabilistic approach that is able to learn the composition structure of the composite services and compute the trust scores for the constituents is advanced.

Another statistical method, the Beta Mixture [51], was employed in [24] to assign trust for the components of the composite services. Trust is assigned for components based on their responsibilities, while taking into consideration the dynamism in the QoS and the fact that not all the observations can be noticed.

Although statistics-based models account for the dynamic characteristics of the QoS parameters, they cannot provide decisive solutions for this problem. In fact, these models suggest tracking the most recent behaviors of the Web services to predict their current performance. Nonetheless, the QoS of the services may change on demand (not in a regular manner) [47], which makes tracking the most recent behavior incapable of making reliable predictions. For instance, an online car rental service may face important degradation in its performance during the promotion time due to the pointedly increased number of orders. In this case, the current performance is unlikely to be predicted from the recent performance since the change in the QoS does not happen in a regular manner. Therefore, a monitoring mechanism that can capture the variations in the performance is recommended [47]. Moreover, these models ignore the collusion scenarios that may occur among the constituents of the composite service and that may lead to false estimations of these constituents' trust values. For instance, constituents may collude according to different scenarios to mislead the predictions. Additionally, statistics-based models do not study the collaboration and task allocation issues among the constituents. Furthermore, the topic of malicious constituents that join compositions to perform malicious objectives was not addressed yet.

4.2.2. Game-theoretic-based models

Game theory is a formal study of conflict and cooperation that applies whenever the actions of several agents are interdependent. Few game-theoretic-based models [27, 26] were proposed to address trust and reputation in the composite architecture. The objective of these models is to model the competition among constituents seeking to get allocated with tasks in the compositions and select hence the appropriate candidate with the aim of maximizing the probability of performing the allocated tasks successfully.

As an example, Yahyaoui [27] proposed a trust-based game whose objective is to model the competition among services seeking to get allocated with tasks and select hence the appropriate candidate. To achieve this, Web services use a Bayesian model to compute a trust value for every other service willing to collaborate with and play a game to select the appropriate candidates.

Game-theoretic-based models address an important topic in the context of composite Web services, which is the task allocation regulation. This issue is important since it helps increase the probability of

Table 4: Comparison summary between the main trust and reputation approaches in the composite architecture

Approach	Model	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
Mehdi et al. [23]	Statistical	✓	✓				✓	✓	✓	✓				
Hang et al. [24]	Statistical	✓	✓			✓	✓	✓	✓	✓	✓			
Paradesi et al. [25]	Statistical	✓	✓	✓			✓	✓	✓					
Yahyaoui [26]	Game theory	✓		✓			✓	✓	✓				✓	
Yahyaoui [27]	Game theory	✓		✓		✓	✓	✓	✓				✓	

the composite service for achieving the allocated task with better performance. However, these models did not capture the whole picture of the task allocation problem. More precisely, they ignore the collusion scenarios that may occur among services. Practically, some services may collude to promote/demote each other or some other Web services, which may lead to inappropriate selection and create unreliable compositions. As in the statistics-based models, game-theoretic-based models ignore as well the topic of malicious constituents that join compositions to perform malicious attacks. Different from statistics-based models, game-theoretic-based models do not evaluate the responsibility of constituents in the composition process.

4.3. Communities of Web Services

Communities of Web services (CWS) can be viewed as groups of services sharing the same functionality but differing in their non-functional properties [3]. Creating communities has a two-fold objective resulting in benefit for both Web services and users. Web services will be exposed to wider groups of users and will have chances to contribute in a greater number of compositions. Users, in their turn, will get their requests fulfilled with better quality as a result of the cooperation that takes place among the services within communities [3]. The topic of trust and reputation has been extensively addressed in the communities of Web services, where the objective is to enable Web services to work and cooperate within a truthful environment. To attain this objective, a collection of requirements have to be satisfied. As communities are composed of single Web services and can involve some kinds of functionally-similar compositions among community members, the requirements proposed for both the single and composite architectures (Table 1 and Table 3 respectively) apply as well for the community-based architecture in addition to other important requirements such as [52, 29, 31, 32, 53, 33, 34, 54]:

- **Criterion** #14: Investigate the community joining strategies in a thoughtful manner, i.e., in a way that enhances/maintains the community's performance and reputation.
- **Criterion** #15: Adapt the trust values to the highly dynamic environment of the communities wherein Web services are continuously joining and leaving.

Table 5: Criteria for the trust and reputation models in the community-based architecture

ID	Criterion	References
C14	Investigate the joining strategies in a thoughtful manner.	[52, 29, 32]
C15	Consider the highly dynamic environment of the communities.	[29, 32]
C16	Consider the existence of selfish Web services.	[29, 31, 53]
C17	Account for the active malicious Web services.	[33, 34]
C18	Study a fully malicious model.	[54]

- **Criterion** #16: Consider the existence of *selfish* or *passive malicious Web services* in the communities whose objective is to manipulate the reputation values by means of malicious actions.

- **Criterion** #17: Consider the existence of *active malicious Web services* whose objective is to join communities to lunch some attacks leading to disrupt the functioning of these communities.

- **Criterion** #18: Study a fully malicious model that mimics the reality, where all the parties that are intelligent agents are assumed to behave maliciously seeking foremost their own objectives.

These criteria are summarized in Table 5. The topic of trust and reputation in the CWS was first addressed in [52], where the authors tried to adapt the architecture of CWS to support trust and reputation models. This was achieved by proposing an architecture of four components: user-agent; provider-agent; extended Universal Description, Discovery and Integration (UDDI)[1]; and reputation system. They defined as well some metrics to evaluate the reputation of the community from the perspectives of both users and providers. Most of the existing trust and reputation models proposed for the CWS build on and extend this reputation model. These models fall into two major classes: analytical models, and game-theoretic-based models. More details about these models and their associated approaches are discussed in the following subsections. Thereafter, the discussed approaches are compared in Table 7 w.r.t the criteria presented in Table 5.

4.3.1. Analytical Models

Analytical models are mathematical models that use equations to analyze the relationships among a set of variables. These models have been used for the CWS to analyze the relationships among the reputation parameters of the Web services in order to help them decide wether to join communities or to work alone.

In [28], the authors perform an analysis on the incentives that would motivate a community (containing one or more elements) of Web services to join another community or to stay alone. For this purpose, they formulate a performance function composed of two factors: use of allocated Web services, and simultaneous obtained feedback. Based on the proposed function, the authors stated that a community will be encouraged

[1] UDDI is a platform-independent XML-based mechanism to register and find Web service applications

to join another community if: (1) it is overloaded by a huge number of requests, or (2) it is unable to attract enough services satisfying its Web services.

In [30], the authors analyze the impacts that reputation parameters have on each other in order to help Web Services decide whether to join community or stay alone. Two cases are considered: Web service is overloaded and Web service is idle. In the first case, the analysis results show that (1) the large increase in the number of requests would result in a decrease in the Web service's reputation, and (2) the change in the reputation in the current time either positively or negatively leads to a negative change in the reputation in the next time unit. In the second case, the analysis revealed that a positive rate of reputation change at a certain time results in a positive rate of change in the next time slot.

In [29], the authors developed an analytical model that analyzes the incentives that would demotivate the community coordinator from behaving maliciously by either increasing its reputation level or decreasing other communities' reputation levels illegally. To tackle this issue, a third-party called agent controller is assigned the role of recognizing the misbehaviors by comparing the community's reputation change (improvement or degradation) between two slots of time and matching this change with a predefined threshold.

Although analytical models tend to provide strong solutions since they are based on mathematical proofs, these models fail to provide solid decision making frameworks for the Web services since they restrict the analysis to few parameters. For example, [28] restricts the reputation assessment to three metrics; thus ignoring some important factors such as capacity of handling requests. Similarly, the analysis presented in [30] is limited to two reputation parameters computed by Web services; thus eliminating the reputation parameters related to the users. Likewise, the authors in [29] limit the analysis to three reputation metrics. Moreover, analytical models provide no bootstrapping mechanism to compute initial trust values for the new Web services and communities. Furthermore, they do not account for the malicious Web services that join communities to launch attacks deteriorating communities' QoS and reputations.

4.3.2. Game-theoretic-based Models

Game theoretical models have been widely investigated in the community-based architecture, where they are mainly used to address the shortcomings of the analytical models and tackle the concept of joining communities in a more systematic manner.

A one-stage game theoretical model has been developed in [32] to provide Web services with a decision making framework that helps them adopt strategies inside and outside communities. Using the proposed

Table 6: Comparison summary among the class models in each architecture

Architecture	Model	Purpose	Limitations
Single	Feedback-based	Build trust value from users' reviews	Unfair ratings. Dependency on the credibility of the majority of raters. Provide no bootstrapping mechanism.
	Statistics-based	Combine different sources of trust	Cannot compute trust values for the new Web services.
	Fuzzy-logic-based	Infer the rationale behind users' reviews	Provide no mechanism to compute the final trust value. Provide no bootstrapping mechanism. Do not consider the trust dynamism.
	Data-mining-based	Analyze users' reviews	Lack of in-depth methodology. Unfair ratings. Provide no bootstrapping mechanism. Do not consider the trust dynamism. Ignore the objective sources of trust.
Composite	Statistics-based	Learn the responsibility of the composite service's constituents	Cannot obtain reliable predictions on the variations in the QoS parameters. Ignore the collusion scenarios among the composite service's constituents. Do not consider the malicious constituents that join compositions to perform malicious attacks. Do not study the collaboration and task allocation issues among constituents.
	Game-theoretic-based	Regulate the task allocation among composite service's constituents	Ignore the collusion scenarios among the composite service's constituents. Do not consider the malicious constituents that join compositions to perform malicious attacks. Do not evaluate the responsibility of constituents in the composition process.
Community	Analytical	Analyze the relationships among the reputation parameters	Provide no bootstrapping mechanism. Limited to few reputation parameters. Do not account for the malicious services that join communities to launch attacks.
	Game-theoretic-based	Provide decision making frameworks for Web services and communities	Rely on fully-honest or semi-honest adversary models. Provide no bootstrapping mechanism.

game, the authors derive a threshold to be compared with the expected performance. If the expected performance exceeds the threshold, then the strategy will be joining for the single Web service and accepting the invitation to join for the community. Another threshold is derived to control the strategies of the Web services inside the communities. If the expected performance exceeds this threshold, the strategy of the single Web service would be leaving the community; otherwise it would prefer to remain. Game theoretical models were used as well to model the collusion scenarios that occur among Web services acting as intelligent agents. The objective is to guarantee a truthful environment where involving entities act honestly. In this context, a repeated game model was derived in [31] in order to maintain sound reputation mechanism in the presence of malicious services seeking to enhance their reputations by means of fake feedback. To this end, the authors discussed four scenarios the controller of the community (charged of monitoring the feedback file against manipulations) may face such as: malicious act not penalized, truthful act penalized, truthful act not penalized, and malicious act penalized. Thereafter, a repeated game of two players (Web service and controller) is analyzed to derive the best strategy for both players. This analysis revealed that if the service

Table 7: Comparison summary between the main trust and reputation approaches in the community-based architecture

Approach	Model	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18
Khosravifar et al. [28]	Analytical	✓	✓				✓	✓	✓						✓	✓			
Khosravifar et al. [29]	Analytical	✓	✓	✓	✓		✓	✓	✓	✓			✓		✓	✓	✓		
Khosravifar et al. [30]	Analytical	✓					✓	✓	✓						✓	✓			
Bentahar et al. [31]	Game theory	✓	✓	✓	✓		✓	✓	✓	✓		✓	✓		✓	✓	✓		
Khosravifar et al. [32]	Game theory	✓	✓				✓	✓	✓				✓		✓	✓			

is made aware of the penalties that it may undergo as well as of the controller's detection accuracy, then the system will fulfill sound and secure state.

Game-theoretic-based models introduced in-depth reasoning about the behaviors and actions of the different agents involved in the community-based architecture and are able hence to provide effective and powerful decision making frameworks for these agents. Nonetheless, the main problem of the game-theoretic-based models in this architecture is that they rely on fully-honest or semi-honest adversary models that assume the existence of one or more trusted parties. For example, the work presented in [32] does not consider the possible malicious nature of the services joining the communities. It assumes hence that all the parties involved in the game (coordinator, single Web services, and users) are trusted. Moreover, the controller agent in [31] is responsible for supervising the feedback file against false feedbacks without considering the case where the controller agent may be itself involved in the collusion between Web services and consumers. Moreover, game-theoretic-based models provide no bootstrapping mechanism to compute initial trust values for the new services and communities.

4.4. Summary of Findings

Table 6 provides a comparison summary among the classes of models defined in each architecture. The table illustrates the purpose behind using each class in the architecture in question and highlights its main limitations. Moreover, we summarize in Table 8 the discussed trust and reputation approaches to help readers visualize and understand them.

5. Discussions and Research Directions

A collection of trust and reputation models has been introduced in the domain of Web services. These models differ in the topics they address, which are imposed mainly by the architecture of Web services in question. They differ as well in the manner they use to construct the trust value for the Web services. Therefore, we base our classification of the existing trust and reputation models on these two perspectives. In fact, we present a two-level classification scheme that classifies the trust and reputation models based on the (1) architecture they are targeting as a high-level classification; and (2) technique they use to construct

Table 8: Summary of the main trust and reputation approaches proposed for Web services

Approach	Addressed Problem	Contribution
Maximilien and Singh [17]	Trust-based selection for Web services	Ontology-based framework that considers user preferences, providers advertisements, and QoS monitoring.
Malik and Bouguettaya [19]	Assessing the reputation of Web service providers	Reputation assessment framework that considers raters credibility, personalized preferences, temporal sensitivity, and first-hand knowledge.
Nguyen et al. [20]	Combining different sources of trust	Trust and reputation model that integrates different kinds of trust sources and evaluates the credibility of raters.
Sherchan et al. [21]	Infer the rationale behind users's reviews	Investigate the relationship between the objective dimension and the subjective dimension using fuzzy approach.
Thurow and Delano [22]	Analyze users' reviews	Text mining technique that extracts information about Web services' QoS parameters from the users' reviews.
Mehdi et al. [23]	Assessing the trustworthiness of composite Web service components	Probabilistic approach that learns the composition structure and computes trust.
Hang et al. [24]	Assigning trust for composite Web service components	Trust-based approach that dynamically learns the responsibilities of components and computes trust.
Yahyaoui [27]	Collaboration among Web services	Trust-based game that models the competition among Web services for tasks allocation.
Elnaffar et al. [52]	Assessing Web services communities using reputation-based approach	Extension of the Web services architecture to support communities and reputation model design.
Khosravifar et al. [28]	Analyzing incentives that encourage Web services to join communities	Performance function formulation to help communities adjust their joining strategies.
Khosravifar et al. [29]	Evaluate the reputation of communities in the presence of malicious coordinators	Sound logging mechanism that motivates the well-behavior of community coordinators.
Khosravifar et al. [30]	Analyzing the impacts that reputation parameters have on each others	Theoretical analysis that helps Web services decide whether to work alone or to join communities.
Bentahar et al. [31]	Maintaining sound reputation in the presence of malicious Web services	Game theoretical model that investigates the incentives that would encourage Web services to act truthfully.
Khosravifar et al. [32]	Help Web services adopt strategies inside and outside communities	Game model between Web services and coordinator to analyze the payoffs for both parties based on different strategies.

the trust value as a low-level classification. Profound analyses and comparisons are derived from these classifications; uncovering prospective topics for future study and investigation. In the following, we discuss the results obtained from these comparisons, highlight some possible research topics in each architecture, and illustrate the future perspectives that are entailed by our work.

5.1. Single Architecture

Trust and reputation mechanisms have been widely used and investigated in the single architecture of Web services. Different approaches were proposed targeting numerous topics. Since each approach focuses on a specific perspective, some important criteria are missed. Practically, some approaches focus on the subjective perspectives and ignore the objective perspectives. Some approaches do not account for the dynamism of the trust. Additionally, some approaches disregard the bootstrapping issue, which constitutes an important challenge for any trust and reputation mechanism. Some proposals don't assess the credibility of the ratings used to build the trust and reputation model, which may trigger collusion and deception problems. Besides, some approaches are based on the assumption that the majority of the ratings are truthful, which is not always realistic. Therefore, a more comprehensive trust and reputation model considering all the mentioned criteria is needed.

5.2. Composite Architecture

Numerous approaches were proposed to tackle trust and reputation in the composite architecture. These approaches are either statistics-based or game-theoretic-based. The goal of the statistics-based models is to learn the responsibility of the composite service's constituents in order to enhance the current compositions and facilitate future selections. The main problem of these models is that they predict the change in performance of the services based on the most recent performance, which cannot yield accurate predictions. It would be recommended to investigate a monitoring mechanism that is able to capture the variations in the QoS parameters of the constituents. Moreover, statistics-based models do not address the task allocation among composite services' constituents and do not consider as well the collusion scenarios that may take place among these constituents and that may influence the predictions. It would be interesting to develop a more comprehensive approach that is able to learn the responsibilities of the constituents based on a monitoring mechanism that captures the dynamism in the performance and under a colluding scenario. On the other hand, game-theoretic-based models focus on the topics of collaboration and task allocation among the constituents of the composite service. Similar to the statistics-based models, game-theoretic-based models ignore the collusion scenarios that may be initiated by the Web services. More specifically, Web services may collude to promote each other and get higher chances to get allocated with tasks and/or promote/demote other services. Thus, it is important to consider the collusion scenarios to obtain fair and reliable task allocations. Furthermore, the topic of active malicious constituents that join compositions to perform malicious objectives is not addressed yet. These malicious constituents may take advantage of several vulnerabilities that exist in the composite architecture to perform their goals such as: long-term partnerships and services' resource constraints. The main attacks that can be launched against the composite architecture are merged and presented with those of community-based architecture in Section 5.3 as the same attacks apply for both architectures since communities can be viewed as long-term compositions among Web services sharing the same functionality. The studied attacks are limited to those that have major impacts on the reputation and QoS of the composite services, associated with the main metrics that influence the trust value assigned by users towards composite Web services. Moreover, the simulation results that show the impact of malicious Web services that launch these attacks on the composite services can be found in Appendix A.

5.3. Community-based Architecture

Trust and reputation models in the community-based architecture fall into two main classes: analytical and game-theoretic-based. The aim of the analytical models is to analyze the relationships among the

reputation parameters of the Web services in order to help them choose strategies either to join communities or to stay alone. The problem of these models is that they are limited to few reputation parameters. More thorough analysis involving a wider set of important parameters is required to provide efficient decision making frameworks. On the other hand, game-theoretic-based models provide more efficient decision making frameworks for the Web services and have the advantage of considering the existence of *malicious agents* that constitute a serious challenge to the community-based architecture. These malicious services may, individually or as a result of collusion with some customers or communities, join the communities for the purpose of launching attacks leading to harm or deteriorate some other community members or the community as a whole. In addition to the vulnerabilities of the composite architecture mentioned in Section 5.2 that are applicable also in the community-based architecture, malicious services may exploit additional vulnerabilities specific to the community-based architecture such as: dynamic topology (freedom to join and leave communities), scalability (no restriction on the number of community members).

Some existing game-theoretic-based models [29, 31] tackled the existence of *passive malicious services* whose objective is to increase their reputations among other members. These approaches fail to provide strong protection against such a misbehavior since they rely on the existence of a central party such as controller agent that will monitor and take decisions. Nonetheless, these parties are intelligent agents that may be tempted to get involved in the collusion scenarios, which may lead to false decisions. A more nested scenario where all the parties are assumed to behave maliciously is recommended. In addition, the topic of *active malicious services* whose objective is to harm or destroy other members or communities by launching several attacks was ignored. In the following, we highlight the main attacks that are applicable on both the composite and community-based architecture [33, 34]. Recall that the studied attacks are restricted to those that are significantly affecting the trust, reputation, and QoS of the Web services compositions/communities.

1. **Request Drop Attack:** Composite Web services and communities are usually based on the assumption that single Web services are willing to cooperate in order to respond to the complex requests with better performance. However, some malicious services may join a certain composition/community and refuse to cooperate and fulfill the requests. The simplest form of this attack is when a certain component/member refuses all the requests it receives. However, this malicious component/member faces the risk of being easily detected and fired by the composition designer or community coordinator. A more intelligent derivation of this attack is when malicious components/members perform

the requests dropping in a selective manner. In such a way, these components/members will drop the requests coming from certain clients or Web services, every t slots of time, or every r requests. This kind of attacks is called *Selective Request Drop* (SRD) attack.

2. **Denial of Service (DoS) Attack:** This attack aims at deteriorating or reducing the service's availability. In this attack, a malicious component/member working within a composition/community may send a request to its partners to exhaust their resources (e.g, memory capacity) in a way that makes them unavailable for responding to further requests. The most two important DoS attacks on XML-based services such as Web services are *Coercive Parsing* and *Oversize Payload* [55]. In the Coercive Parsing attack, a pointedly nested XML document is used to consume the service's memory. In the Oversize Payload attack, an extremely large XML document is employed for this purpose.
3. **Sybil Attack:** This attack takes place when malicious components/members create illegitimately a large number of fake identities (fabricated identities) or impersonate other legitimate Web services in the composition/community (stolen identities). The goal of the attacker in this case is to appear and operate as multiple distinct Web services in a way that enables it to take control over the whole composition/community. This attack may occur only in case of communities and long-term compositions and may be exploited by the attackers to achieve several malignant objectives in different aspects.
4. **Outage Attack:** This attack occurs when malicious components/members committed to perform a certain task within a whole process suddenly stop their functioning, which leads to interrupt the functioning of the whole process.
5. **Sinkhole Attack:** In this attack, malicious components/members seek to lure nearly all the requests (from clients/or from other services). This attack is done by making a compromised component/member look attractive to clients/Web services by claiming bogus reputation.
6. **Eavesdropping Attack:** This attack happens when malicious components/members collect information from the composition/community (e.g., application-specific messages content) they belong to in favor of other competitor compositions/communities. Thereafter, these malicious components/members may decrease their performance in a way that ends them up being fired from the current composition/community. This allows them to join other compositions/communities and use the collected information for malicious purposes.
7. **Composition/Community Exclusion Attack:** In this attack, malicious components/members deteriorate the reputation of the composition/community they belong to in a way that makes this com-

position/community be undesirable to deal with by any client or service. This attack can be performed by applying the request drop, DoS, outage, or sybil attacks in a way that makes the composition/community look unable or unwilling to fulfill the incoming requests.

8. **Component/Member Exclusion Attack:** This attack happens when malicious components/members start launching attacks (e.g., DoS) leading to exclude a specific victim from the composition/community by decreasing its reputation in a drastic manner.

The simulation results that show the impact of active malicious Web services launching each of these attacks on the community-based architecture can be found in Appendix A.

5.4. Future Perspectives

This work classifies and compares trust and reputation models proposed for Web services based on a set of defined criteria. The results of the work may be used in the benefit of Web services' providers, consumers, and research community. By developing a classification scheme and proposing a set of criteria for each class, we aim to help providers enhance the quality of their services by letting them learn the factors that affect the user's judgement on the service. From criteria $C1$, $C2$, and $C6$ (Table 1), providers will learn that users care about a wide variety of QoS metrics when building their reputation towards services and that they should keep up the quality of services at a good level since the trust is subject to change over the time. Criteria $C8 - C13$ (Table 3) help providers enhance the quality of their compositions by stressing the importance of the issues of learning the performance of the composite service's constituents and managing the task allocation in a thoughtful manner. Criteria $C14 - C18$ (Table 5) help providers design high-quality and secure communities of Web services. As a result, consumers will enjoy services with better quality and performance. They will be motivated as well to provide truthful feedback by learning from criteria $C3$, $C4$, and $C7$ (Table 1) that the reputation system should be able to discard untrustworthy ratings. Moreover, our analysis reveals some topics that are interesting to investigate in the domain of Web services. More specifically, we raise the topic of active malicious Web services in the composite and community-based architectures by defining such malicious services, clarifying their objectives, highlighting some vulnerabilities that they may exploit, and elucidating their negative impacts by means of simulation experiments conducted on a real-life dataset. Thus, our work may be used by Web services' security research community as a starting point to study and explore security-based models targeting these malicious services.

6. Conclusion

In this survey, we presented a two-level classification scheme that classifies the trust and reputation models proposed for Web services according to the architecture they target, and the technique they use to build the trust value. A collection of criteria were defined for each architecture based on which the class models as well as the major approaches in each architecture are compared. This comparison reveals some important issues that need further study and investigation. One of the important challenges is the existence of active malicious Web services in the composite and community-based architectures and whose objective is to harm the compositions/communities by decreasing their performance and reputation. In this context, we described eight possible attacks that have major impacts on compositions/communities' performance and reputation. We conducted as well simulations on real-life dataset to show the negative impacts of such malicious services using several QoS metrics. Simulation results reveal that the existence of malicious services considerably increases the response time and decreases the reputation, availability, and throughput. This opens the door for future researchers to investigate security-based models for the purpose of protecting the composite and community-based architectures from such potential attacks.

References

[1] G. Alonso, F. Casati, H. Kuno, V. Machiraju, Web Services: Concepts, Architectures and Applications, 1st Edition, Springer Publishing Company, 2004.

[2] J. Pathak, S. Basu, V. Honavar, Assembling composite web services from autonomous components, in: Proceedings of the Conference on Emerging Artificial Intelligence Applications in Computer Engineering, IOS Press, 2007, pp. 394–405.

[3] Z. Maamar, S. Subramanian, P. Thiran, D. Benslimane, J. Bentahar, An approach to engineer communities of web services: concepts, architecture, operation, and deployment, International Journal of E-Business Research 5 (4) (2009) 1–21.

[4] O. A. Wahab, H. Otrok, A. Mourad, VANET QoS-OLSR: QoS-based clustering protocol for Vehicular Ad hoc Networks, Computer Communications 36 (13) (2013) 1422–1435.

[5] K. Voevodski, M.-F. Balcan, H. Rglin, S.-H. Teng, Y. Xia, Active clustering of biological sequences, Journal of Machine Learning Research 13 (1) (2012) 203–225.

[6] K. de Valck, G. H. van Bruggen, B. Wierenga, Virtual communities: A marketing perspective, Decision Support Systems 47 (3) (2009) 185–203.

[7] A. Jsang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, Decision Support Systems 43 (2) (2007) 618–644.

[8] D. Gambetta, Trust: making and breaking cooperative relations, Oxford Basil Blackwell, 1988.

[9] A. A. Selcuk, E. Uzun, M. R. Pariente, A review on computational trust models for multi-agent systems, International Journal of Network Security 6 (3) (2004) 235–245.

[10] O. A. Wahab, H. Otrok, A. Mourad, A Dempster-Shafer Based Tit-for-Tat Strategy to Regulate the Cooperation in VANET Using QoS-OLSR Protocol, Wireless Personal Communications 75 (3) (2014) 1635–1667.

[11] O. A. Wahab, H. Otrok, A. Mourad, A cooperative watchdog model based on Dempster-Shafer for detecting misbehaving vehicles, Computer Communications 41 (2014) 43–54.

[12] Z. Lin, D. Li, B. Janamanchi, W. Huang, Reputation distribution and consumer-to-consumer online auction market structure: an exploratory study, Decision Support Systems 41 (2) (2009) 435–448.

[13] Y. Wang, J. Vassileva, Toward trust and reputation based web service selection: A survey, International Transactions on Systems Science and Applications (ITSSA) Journal 3 (2) (2007) 118–132.

[14] V. Mareeswari, D. E. Sathiyamoorthy, A survey on trust in semantic web services, International Journal of Scientific & Engineering Research 3 (2) (2012) 1–5.

[15] S. Phoomvuthisarn, A survey study on reputation-based trust mechanisms in service-oriented computing, Journal of Information Science and Technoogy 2 (2) (2011) 1–12.

[16] N. Dragoni, A survey on trust-based web service provision approaches, in: Third International Conference on Dependability (DEPEND), IEEE, 2010, pp. 83–91.

[17] E. M. Maximilien, M. P. Singh, Multiagent system for dynamic web services selection, in: Proceedings of 1st Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE at AAMAS), 2005, pp. 25–29.

[18] Z. Malik, A. Bouguettaya, Rater credibility assessment in web services interactions, World Wide Web 12 (1) (2009) 3–25.

[19] Z. Malik, A. Bouguettaya, Rateweb: Reputation assessment for trust establishment among web services, VLDB Journal 18 (4) (2009) 885–911.

[20] H. T. Nguyen, W. Zhao, J. Yang, A trust and reputation model based on bayesian network for web services, in: IEEE International Conference on Web Services, IEEE, 2010, pp. 251–258.

[21] W. Sherchan, S. W. Loke, S. Krishnaswamy, A fuzzy model for reasoning about reputation in web services, in: Proceedings of the ACM Symposium on Applied Computing, SAC, ACM, 2006, pp. 1886–1892.

[22] N. A. Thurow, J. D. Delano, Selection of web services based on opinion mining of free-text user reviews, in: Proceedings of the International Conference on Information Systems, Association for Information Systems, 2010, pp. 42–51.

[23] M. Mehdi, N. Bouguila, J. Bentahar, A QoS-based trust approach for service selection and composition via Bayesian networks, in: IEEE 20th International Conference on Web Services, IEEE, 2013, pp. 211–218.

[24] C.-W. Hang, A. K. Kalia, M. P. Singh, Behind the curtain: Service selection via trust in composite services, in: IEEE 19th International Conference on Web Services, IEEE, 2012, pp. 9–16.

[25] S. Paradesi, P. Doshi, S. Swaika, Integrating behavioral trust in web service compositions, in: IEEE 19th International Conference on Web Services, IEEE, 2009, pp. 453–460.

[26] H. Yahyaoui, Trust assessment for web services collaboration, in: IEEE International Conference on Web Services, IEEE, 2010, pp. 315–320.

[27] H. Yahyaoui, A trust-based game theoretical model for web services collaboration, Knowledge-Based Systems 27 (2012) 162–169.

[28] B. Khosravifar, J. Bentahar, A. Moazin, Z. Maamar, P. Thiran, Analyzing communities vs. single agent-based web services: Trust perspectives, in: IEEE International Conference on Services Computing (SCC), IEEE, 2010, pp. 194–201.

[29] B. Khosravifar, J. Bentahar, A. Moazin, P. Thiran, Analyzing communities of web services using incentives, International Journal of Web Services Research 7 (3) (2010) 30–51.

[30] B. Khosravifar, J. Bentahar, A. Moazin, Analyzing the relationships between some parameters of web services reputation, in: IEEE 19th International Conference on Web Services, 2010, pp. 329–336.

[31] J. Bentahar, B. Khosravifar, M. A. Serhani, M. Alishahia, On the analysis of reputation for agent-based web services, Expert Systems with Applications 39 (16) (2012) 12438–12450.

[32] B. Khosravifar, J. Bentahar, R. Mizouni, H. Otrok, M. Alishahi, P. Thiran, Agent-based game-theoretic model for collaborative web services: Decision making analysis, Expert Systems with Applications 40 (8) (2013) 3207–3219.

[33] C. Karlof, D. Wagner, Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures, Ad Hoc Networks 1 (2) (2003) 293–315.

[34] B. Wu, J. Chen, J. Wu, M. Cardei, A survey of attacks and countermeasures in mobile ad hoc networks, in: Wireless Network Security, Springer, 2007, pp. 103–135.

[35] S. Vavilis, M. Petkovi, N. Zannone, A reference model for reputation systems, Decision Support Systems 61 (2014) 147–154.

[36] L. Liu, M. Munro, Systematic analysis of centralized online reputation systems, Decision Support Systems 52 (2) (2012) 438–449.

[37] Z. Malik, A. Bouguettaya, Reputation bootstrapping for trust establishment among web services, IEEE Internet Computing 13 (1) (2009) 1089–7801.

[38] J. H. Abawajy, A. M. Goscinski, Computational Science ICCS, Vol. 3994 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, Ch. A Reputation-Based Grid Information Service, pp. 1015–1022.

[39] E. M. Maximilien, M. P. Singh, Conceptual model of web service reputation, SIGMOD Record 31 (4) (2002) 36–41.

[40] H.-H. Li, X.-Y. Du, X. Tian, A review-based reputation evaluation approach for web services, Journal of Computer Science and Technology 24 (5) (2009) 893–900.

[41] L. Wasserman, All of Statistics: A Concise Course in Statistical Inference, Springer Publishing Company, Incorporated, 2010.

[42] P. K. Atrey, M. A. Hossain, A. E. Saddik, Association-based dynamic computation of reputation in web services, International Journal of Web and Grid Services 4 (2) (2008) 169–188.

[43] G. Bojadziev, M. Bojadziev, Fuzzy Sets, Fuzzy Logic, Applications, Vol. 5, World Scientific, 1996.

[44] S. Nepal, W. Sherchan, J. Hunklinger, A. Bouguettaya, A fuzzy trust management framework for service web, in: IEEE International Conference on Web Services, 2010, pp. 321–328.

[45] F. V. Jensen, T. D. Nielsen, Data Mining: Concepts and Techniques, 3rd Edition, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, 2011.

[46] O. A. Wahab, M. O. Hachami, A. Zaffari, M. Vivas, G. G. Dagher, DARM: a privacy-preserving approach for distributed association rules mining on horizontally-partitioned data, in: Proceedings of the 18th International Database Engineering & Applications Symposium, ACM, 2014, pp. 1–8.

[47] T. Zhang, J. Ma, C. Sun, Q. Li, N. Xi, Service composition in multi-domain environment under time constraint, in: IEEE International Conference on Web Services, IEEE, 2013, pp. 227–234.

[48] F. Skopik, D. Schall, S. Dustdar, Modeling and mining of dynamic trust in complex service-oriented systems, Information Systems 35 (7) (2010) 735–757.

[49] M. Mehdi, N. Bouguila, J. Bentahar, Trustworthy web service selection using probabilistic models, in: Proceedings of the IEEE 19th International Conference on Web Services, ICWS, 2012, pp. 17–24.

[50] L. Li, Y. Wang, A subjective probability based deductive approach to global trust evaluation in composite services, in: IEEE International Conference on Web Services, IEEE, 2011, pp. 604–611.

[51] Z. Ma, A. Leijon, Bayesian estimation of beta mixture models with variational inference, IEEE Transactions on Pattern Analysis and Machine Intellelligence 33 (11) (2011) 2160–2173.

[52] S. Elnaffar, Z. Maamar, H. Yahyaoui, J. Bentahar, P. Thiran, Reputation of communities of web services - preliminary investigation, in: 22nd International Conference on Advanced Information Networking and Applications - Workshops, 2008, pp. 1603–1608.

[53] O. A. Wahab, Cooperative clustering models for Vehicular Ad Hoc Networks, Master's thesis, Lebanese American University (2013).

[54] N. Dragoni, A survey on trust-based web service provision approaches, in: Third International Conference on Dependability (DEPEND), IEEE, 2010, pp. 83–91.

[55] N. Gruschka, N. Luttenberger, Security and Privacy in Dynamic Environments, Vol. 201 of IFIP International Federation for Information Processing, Springer US, 2006, Ch. Protecting Web Services from DoS Attacks by SOAP Message Validation, pp. 171–182.

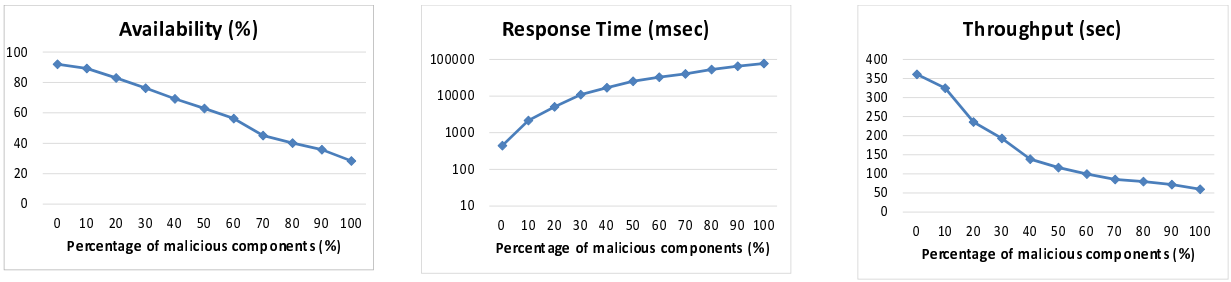
[56] Z. Cao, Eavesdropping or disrupting a communication - on the weakness of quantum communications, IACR Cryptology ePrint Archive (2013) 474–479.

[57] E. Khosrowshahi-Asl, J. Bentahar, H. Otrok, R. Mizouni, Efficient community formation for web services, IEEE Transactions on Services Computing (in press).

Appendix A. Impact of malicious Web services on the composite and community-based architectures

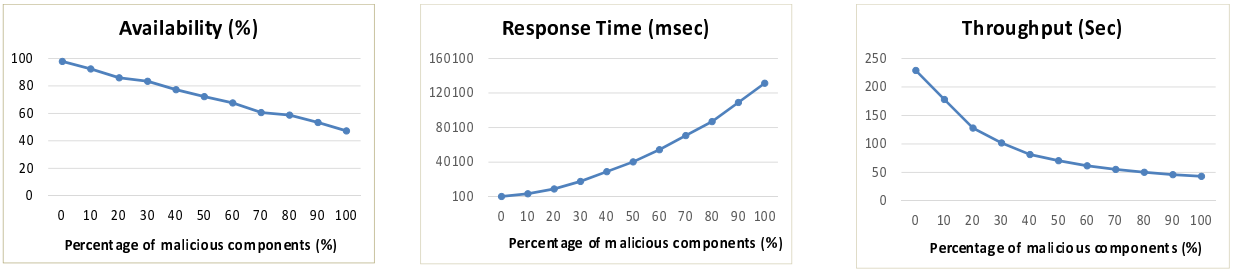
To study the impact of the active malicious Web services, Figures A.2, A.3, and A.4 describe their effects on the composite architecture while Figures A.5, A.6, and A.7 depict their impacts on the community-based architecture. It is well-predictable that the existence of malicious Web services leads to negative implications on the QoS and reputation parameters. However, by advancing simulations on various types of attacks, we are providing readers with the ability to visualize and compare the implications of these attacks. This helps them infer the security plans that should be designed to prevent and/or detect such attacks. For example, one may notice from Figure A.4a that the availability of the composite service begins to drop in a severe manner starting from 10% of Sinkhole attackers. This is due to the fact that although the number of attackers is relatively small, malicious services in this attack work on attracting nearly all the requests incoming to the composition by claiming bogus reputation, which allows them to perform the drop in an extremely severe manner. Similarly, it is worth observing as well that the availability in the Sinkhole attack drops more severely than that of both the Selective Request Drop (Figure A.2a) and DoS (Figure A.3a) attacks. The same intuition applies as well for the community-based architecture. As a result, the reader may conclude that targeting the Sinkhole attack is extremely urgent and that the existence of even a small number of such attackers should not be tolerated.

Several metrics are used throughout simulations such as: reputation (set initially to a value between 0.49 and 0.7 and is updated continuously by the rewards/penalties received by Web services in response to their performance), availability (time period in which a Web service is ready for use), response time (time between the submission of the request and the receipt of the response), and throughput (number of requests that can be processed per time unit). The different attacks described in Section 5.3 are implemented except for the eavesdropping attack, which is a passive attack [56]. In fact, all what malicious services can do in this type of attacks is to monitor and gather information about the compositions/communities to which they belong, which does not have a direct impact on these compositions/communities. However, this attack may be used indirectly in different ways to harm the compositions/communities' reputation and performance. The simulation application is written in C# using Visual Studio and the domain of flight booking is addressed. The information related to Web services is populated from a real dataset that includes 2507 real services functioning on the Web and containing the QoS values of 9 parameters [57]. Each user's request contains the flight dates, the origin and destination, type of tickets (one way or return), and number of guests. The response contains different flights with different companies, prices, timing, etc.



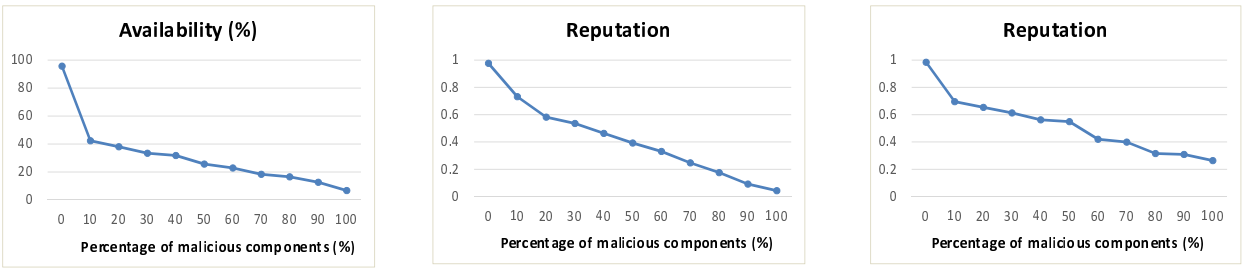
(a) SRD decreases the availability (b) SRD increases the response time (c) SRD decreases the throughput

Figure A.2: Impact of selective request drop attack on the composite architecture



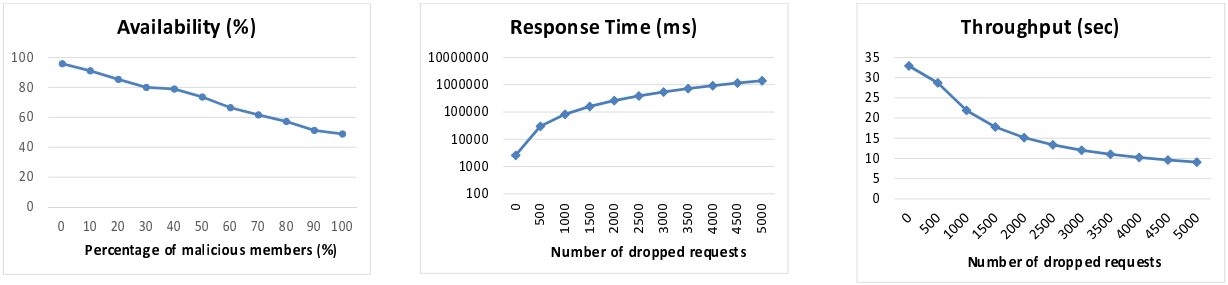
(a) DoS decreases the availability (b) Outage increases the response time (c) Sybil decreases the throughput

Figure A.3: Impact of DoS, Outage, and Sybil attacks on the composite architecture



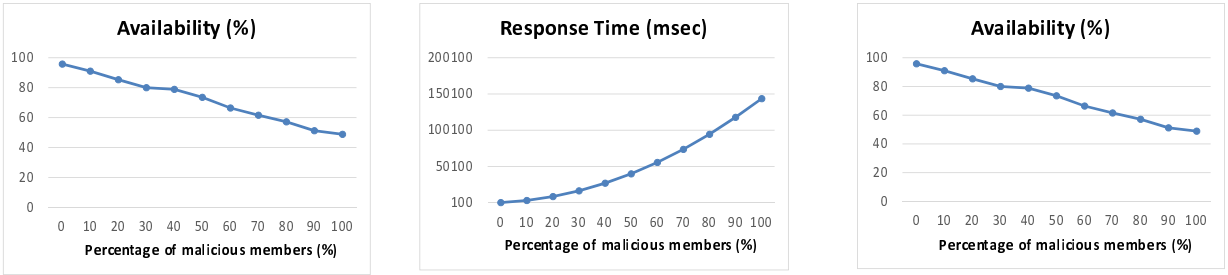
(a) Sinkhole decreases the availability (b) Composition Exclusion deceases the reputation (c) Component Exclusion deceases the reputation

Figure A.4: Impact of Sinkhole, Composition Exclusion, and Component Exclusion attacks on the composite architecture



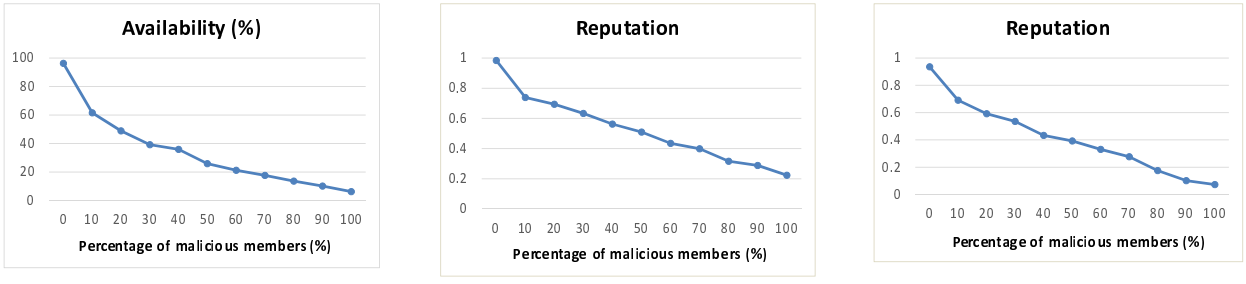
(a) SRD decreases the availability (b) SRD increases the response time (c) SRD decreases the throughput

Figure A.5: Impact of selective request drop attack on the community-based architecture



(a) DoS decreases the availability (b) Outage increases the response time (c) Sybil decreases the throughput

Figure A.6: Impact of DoS, Outage, and Sybil attacks on the community-based architecture



(a) Sinkhole decreases the availability (b) Community Exclusion deceases the reputation (c) Member Exclusion deceases the reputation

Figure A.7: Impact of Sinkhole, Community Exclusion, and Member Exclusion attacks on the community-based architecture