International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 1.982| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 2, Issue 5, September-October 2015 ||

Evaluation of REST Web Service Descriptions for Graph-based Service Discovery with a Hypermedia Focus

Sakshi Sharma[1*], Natasha Dutta[2]

Salesforce Administrator Jul, KONI Hospitality, Chula Vista, CA[1*]

Project Engineer, Online Micro Services, India[2]

ABSTRACT: Service-oriented architecture, or SOA, is a model that encourages collaboration between services that perform different but related tasks. With the advent of online service technology, microservice architecture—a key element of modern online applications that evolved from SOA—became feasible. As the number of self-contained services has grown, it has become more difficult to find the finest one. People struggle to choose the finest service because so many of them perform comparable tasks. Providing customers with the best service as soon as possible is essential to maintaining an efficient infrastructure because unknown services could increase ecosystem expenses. Techniques such as ontology-based, syntactic, and semantic-aware approaches have been proposed to raise the standard and effectiveness of service discovery algorithms.

KEYWORDS: Service-Oriented Architecture (SOA), Web Service Discovery, SOAP, REST, Microservices, Quality of Service (QoS), API, Data Mining.

I. INTRODUCTION

Service-Oriented Architecture (SOA) is an approach for designing distributed systems that provide application functionality as independent, language, and platform-agnostic services [1]. Within this framework, each service operates independently, performing specific functions. Web Services (WS), which are web-based applications, can be published, distributed, and used independently, thanks to technologies like Extended Markup Language (XML), Simple Object Access Protocol (SOAP), Universal Description, Discovery, and Integration (UDDI), and Web Services Description Language (WSDL). These technologies ensure the convenience and interoperability of data, making web services highly popular. While SOAP is primarily used for data communication, XML systematically defines data using well-defined labels [2]. However, the growing number of web services, driven by their many advantages, has made the process of discovering suitable services increasingly complex.

SOAP, a key communication protocol within SOA, facilitates messaging and data exchange between services. On the other hand, web services often use REST (Representational State Transfer)-based APIs, which have become a widely adopted architectural style for communication and data sharing in web-based applications. RESTful services utilize the HTTP protocol to offer APIs that represent resources and support CRUD (Create, Read, Update, Delete) operations. Although both SOAP and REST are protocols for service communication within an SOA framework, microservices, which follow the core principles of SOA, typically implement REST-based APIs, distinguishing them from SOA's traditional use of SOAP-based communication protocols [3].

The process of discovering a web service generally involves three key steps: first, providers advertise their web services in public repositories by publishing programming interfaces through description files like WSDL. Then, users submit a web service request, after which a web service discovery model identifies a list of potential service candidates, usually sorted and displayed in a specific order for the user to choose from. The discovery phase is critical because services that aren't discovered cannot be utilized. However, the retrieved list may include several services offering similar functionalities, making Quality of Service (QoS) a decisive factor. QoS information, which includes aspects like performance (e.g., response time, latency), availability, accessibility, and security, plays a crucial role in determining the quality of prospective web services. Despite its importance, QoS information provided by service providers isn't always accurate. Data mining and machine learning techniques are used to improve the web service discovery process [4]. Clustering techniques, for example, reduce the search space by grouping web services based on various similarity criteria, while association rule algorithms help build recommendation models by identifying relationships between services and users.

To develop more effective methods for web service discovery, it's essential to study previous research in this area. Analyzing earlier studies on web service discovery can provide valuable guidance for new approaches and offer insights into overcoming the challenges these methods face. Therefore, this study aims to conduct a Systematic Literature Review (SLR) to comprehensively understand the methodologies used in web service discovery. We focused on examining and analyzing 54 scientific publications from the last decade, particularly those that included experimental results. This SLR specifically concentrates on the discovery of web services within the microservice ecosystem. The collection of papers was compiled from electronic databases over the past ten years using a keyword-based approach to ensure all relevant literature was included. Exclusion criteria guided the selection process, and quality evaluation questions were used to assess the quality of the papers [5]. Although the selection of keywords and databases may have caused us to miss some relevant studies on online service discovery, the collaborative effort of the authors in this comprehensive review enhances its credibility. During the data extraction phase of the SLR, the authors developed a data extraction file, meticulously reviewing and verifying all study questions and responses to ensure that the data was accurately synthesized and interpreted [6]. This study focuses on understanding the process of discovering web services (WS) and aims to develop better methods by reviewing previous research in the field. By examining earlier studies, the researchers hope to gain valuable insights into the different techniques, algorithms, and challenges related to web service discovery. This knowledge can help in creating more effective discovery methods. The main objective of this study is to conduct a Systematic Literature Review (SLR), which involves a detailed analysis of previous research on web service discovery, particularly within the microservice ecosystem. The researchers reviewed 54 scientific publications from the last 10 years, giving priority to those that included experimental results.

The researchers used a keyword-based approach to search for papers in electronic databases to ensure they covered all relevant literature. They applied specific criteria to exclude irrelevant studies and used quality evaluation questions to assess the reliability of the papers they selected. Although the choice of keywords and databases might have led to missing some important studies, the collaborative effort of the authors in reviewing the literature adds credibility to the findings [7]. During the data extraction phase of the SLR, the authors created a data extraction file to systematically review and verify all study questions and responses. This careful approach ensures that the data is accurately interpreted and synthesized, providing a solid foundation for understanding and improving web service discovery methods.

1.1 Role of Hypermedia in RESTful Services

At the heart of RESTful services lies the principle of Hypermedia as the Engine of Application State (HATEOAS). This principle dictates that the interactions within a RESTful application should be navigated and driven by hypermedia links, which are provided by the server in response to client requests. Essentially, hypermedia links act as the connective tissue between different resources, guiding the client through various states and operations within the application without requiring prior knowledge of the service's internal structure.

Hypermedia plays a pivotal role in achieving the flexibility and scalability that RESTful architectures are known for. By embedding links within the representations returned to the client, the server can dynamically adjust the available actions based on the current state of the application. For example, when a client requests information about a specific resource, the server not only returns the resource data but also includes links to related resources or possible actions, such as updating or deleting the resource. This approach allows the client to discover and interact with different parts of the application organically, based solely on the information provided in the responses, rather than relying on hardcoded pathways or predefined workflows.

This hypermedia-driven interaction is key to maintaining loose coupling between the client and the server. In traditional tightly coupled systems, clients are often dependent on a rigid contract or set of rules to interact with the server, making any changes to the server's structure or endpoints potentially disruptive. However, in a RESTful architecture that adheres to HATEOAS, the client does not need to know about the server's internal workings in advance. Instead, it can dynamically adapt to the responses it receives, following the hypermedia links to perform the necessary actions. This decoupling allows for greater flexibility and resilience in the face of changes, as the server can evolve independently of the client, as long as it continues to provide appropriate hypermedia links.

Despite its advantages, effectively leveraging hypermedia for service discovery poses certain challenges. Service discovery in RESTful architectures is not just about locating the service; it's also about understanding how to interact with it and what operations are available at any given moment. Unlike SOAP-based web services, which rely heavily on WSDL (Web Services Description Language) for a detailed, upfront description of available operations, RESTful services rely on hypermedia to convey this information incrementally. This means that to fully harness the power of hypermedia, there must be a robust mechanism in place to describe these RESTful services in a way that clients can easily navigate and understand.

This need for an effective service description mechanism is particularly important in environments where services are numerous, dynamic, and may need to be discovered or integrated on the fly. In such cases, relying solely on the hypermedia links provided during runtime may not be sufficient for clients to efficiently discover and utilize all available services. Instead, a more structured approach to service description—one that still honors the principles of REST and HATEOAS—is required to aid in the discovery and integration process. Various description languages and formats, such as OpenAPI and Hydra, have been developed to address this need, providing a way to predefine the structure and possible interactions of RESTful services while still allowing for the dynamic, hypermedia-driven interactions that are central to REST.

Graph-Based Approaches for Service Discovery

Graph-based approaches to service discovery have emerged as a powerful method for managing the complexity and dynamism of modern web service ecosystems. In traditional service discovery methods, services are often listed or indexed in a linear fashion, which can be limiting when dealing with large-scale, interconnected systems where services are interdependent and constantly evolving. A graph-based approach, however, models services and their interactions as a network of nodes and edges, where each node represents a service, and each edge signifies a relationship or interaction between services. This structure is particularly well-suited to capturing the intricate web of dependencies, relationships, and potential integration paths that characterize complex service environments. Unlike traditional approaches, where services might be discovered in isolation, a graph-based model allows for the exploration of service clusters, where related services are grouped together based on their connections. This clustering makes it easier to identify not only individual services but also the most effective pathways for integrating multiple services to achieve a specific goal. For example, in a microservices architecture, where each service typically performs a distinct function but must work in concert with others, a graph-based approach can help developers or automated systems identify the optimal sequence of service calls to achieve the desired outcome.

When applied to RESTful services, graph-based service discovery becomes even more powerful by leveraging the hypermedia-driven nature of REST. In RESTful architectures, services are often interconnected through hypermedia links, which guide clients from one resource to another. These links can be naturally mapped onto the edges of a graph, while the services or resources themselves become the nodes. As a result, a graph-based discovery mechanism can effectively model the RESTful service ecosystem, capturing the dynamic and evolving nature of these connections. This allows for a more efficient discovery process, where clients or automated systems can explore available services by following the links represented in the graph, much like how they would navigate a RESTful API in practice.

II. LITERATURE REVIEW

This survey examines context-warehousing, user-side, clustering, and recommendation-based online service discovery techniques [9]. This article expands on the current state of clustering and classification approaches [10]. In the survey study of Sagayaraj and Santhoshkumar, the clustering model is structured according to the clustering of web Service discovery methods. This paper examines the drawbacks of web service discovery technologies and provides a summary of the issues posed by various methodologies [11].

Tomas Vitvar (2008) The rise of Web 2.0 has introduced the Programmable Web, where an increasing number of websites offer machine-oriented APIs and web services. However, many of these APIs are described only in text within HTML documents, which limits the ability of developers to effectively use these services due to the lack of machine-readable descriptions. To address this, we propose a microformat called hRESTS (HTML for RESTful Services) designed for machine-readable API descriptions, supported by a straightforward service model. The hRESTS microformat outlines key aspects of services, including operations, inputs, and outputs. We introduce two extensions of hRESTS: SA-REST, which focuses on aspects of public APIs that are particularly relevant for mashup developers, and MicroWSMO, which enhances support for semantic automation.

In practice, REST services are often described through informal, ad-hoc, and semi-structured documents, typically written in natural language, which further exacerbates the problem of tight coupling. The few existing REST service descriptions usually follow an operation-centric approach, but these offer limited benefits to developers and consumers. To address this, we propose a service description model centered on hypermedia, which allows the creation of a graph that captures state transitions within an activity layer. Additionally, we represent the semantics of resources, transitions, and responses within a semantic layer. By utilizing graph queries, this model enables easier service discovery and composition by traversing the graph. Our service model is implemented using Microdata-based annotations and a JSON description format. To demonstrate our approach, we developed a prototype using Neo4J and selected a set of real Web APIs to illustrate its effectiveness.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 1.982| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

IJARETY

|| Volume 2, Issue 5, September-October 2015 ||

The second point focuses on semantic service sections that consider matching. The third point focuses on the pre-processing stage before the pairing stage. The final point focuses on measurement to evaluate techniques [1]. Malaimalavathani and Gowri (2013) evaluate and discuss service discoverability in terms of semantic web service discovery to find the most appropriate service for a given application to use [12]. Phalnikar and Khutade (2012) present a detailed investigation of web service discovery systems utilizing service quality(QoS) factors. It also emphasizes the significance of ontology in representing nonfunctional requirements in web service discovery [5]. Presented a Systematic Literature Review paper on web service clustering approaches aimed at enhancing service discovery, selection, and recommendation [13].

In today's landscape, where mobile devices and web services are prevalent, the choice between SOAP/WSDL and REST is crucial, especially given the physical constraints of mobile devices, such as limited processing power, memory, and intermittent wireless connections. I developed a prototype system and conducted extensive experiments comparing SOAP/WSDL and RESTful web services using the same mobile client. The results clearly show that RESTful web services significantly outperform SOAP/WSDL services in terms of performance and scalability [14].
TAD transforms OpenAPI (Swagger) documents related to RESTful services into a graph structure and automatically annotates the graph nodes with semantic concepts using Latent Dirichlet Allocation (LDA) and WordNet. TAD enables service composition based on user requirements through two key modules: a service discovery chain and a logical-operation-based composition module. The service discovery chain employs the Hungarian algorithm to evaluate the compatibility of service interfaces, helping to find services that can bridge the gap between user requirements and the services discovered. The logical-operation-based composition module identifies services that semantically align with the user's needs, based on the structure of the service flow. These candidate services are then passed to the service discovery chains, facilitating a simultaneous search for potential composition solutions. The system prototype and experimental results demonstrate the effectiveness and practicality of the proposed TAD scheme.

III. METHODOLOGY

This section outlines the procedures for data collection, formulating research questions, and selecting relevant articles. From an initial pool of 764 publications, 54 were selected based on specific criteria. These criteria focused on studies from the last decade related to "web service discovery" and "microservice discovery," using databases such as IEEE Xplore, Science Direct, Springer Link, and Scopus. Key considerations included the availability of full-text access, clarity of language, and inclusion of experimental results, which provide valuable insights into methods, algorithms, and technologies. Inclusion criteria emphasized papers that directly address web service discovery and include experimental data. Exclusion criteria, detailed in Table 2, were developed based on existing guidelines and previous systematic literature reviews. To determine the relevance of papers, each was ranked using a quality score system based on relevance, validity, reliability, and documentation, with scores assigned based on responses to specific questions (Table 4). Low-scoring papers were excluded, resulting in a final selection of 54 papers, primarily sourced from IEEE Xplore.

The synthesis of these papers reveals that research on the web service discovery process is growing, with a distribution of publication types shown in Figures 1 and 2, and a breakdown of sources in Figure 3. This indicates an increasing focus on the critical discovery process for web services in recent research.

Table 1: Identified Research Questions.

RQ	Research questions
RQ1	What method is employed for service discovery?
RQ2	What complementary functions does service discovery serve?
RQ3	What meta-heuristic techniques and algorithms are employed?
RQ4	To what extent is the implementation strategy cloud-based?
RQ5	To what extent is the implementation strategy data or service-centric?
RQ6	What are the challenges, and potential solutions in this domain?

Table 2: Exclusion Criteria

ID	Exclusion criteria
1	The paper is not fully accessible (i.e., papers that include only abstract are excluded).
2	The language of the article is not English.
3	The article does not contain experimental results.

Table 3: The distribution of papers per database.

Database	Number of papers after query search	Selected primary studies
1. Science Direct	99	5
2. IEEE Xplore	394	34
3. Scopus	219	13
4. Springer Link	52	2
Total	764	54

Table 4: Quality Evaluation Questions.

ID	Questions
Q1	Is the purpose of the research specified?
Q2	Are the scope and context of the study clearly defined?
Q3	Are the variables used in the research valid and reliable?
Q4	Is the research process sufficiently documented?
Q5	Are the main findings clearly stated in terms of validity and reliability?

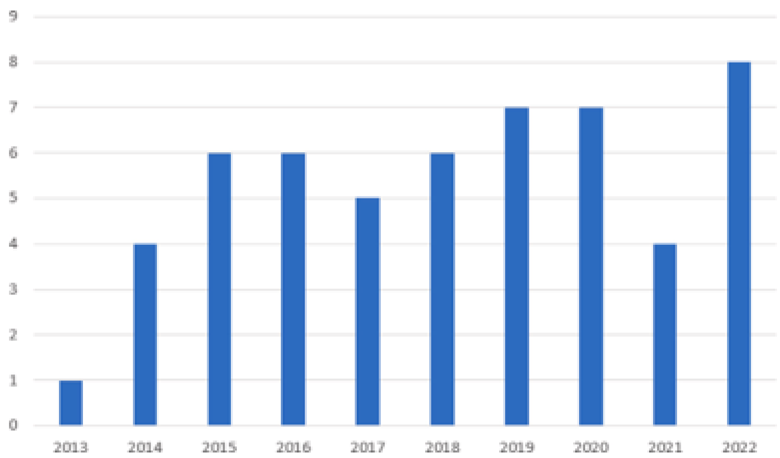

Figure 1: Distribution of papers per year.

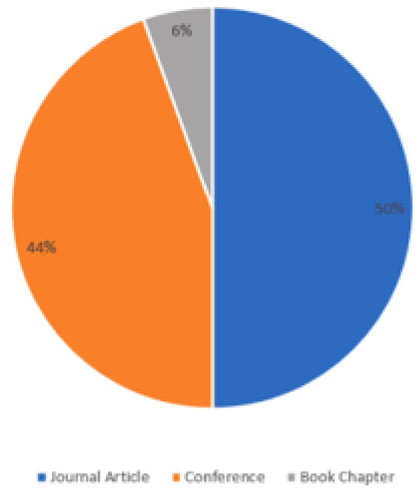

Figure 2: Type of Publications

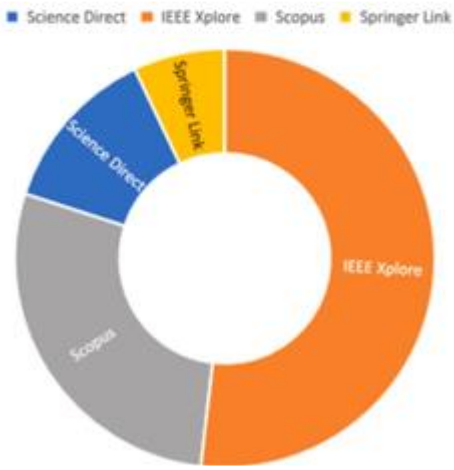

Figure 3: The distribution of studies per search database

IV. RESULT AND DISCUSSION

After designing a form that incorporated the research questions, we conducted a detailed review of each selected publication using this form to systematically evaluate them. Upon completing the review process, we synthesized the data by identifying similarities and differences across the publications. The data were then compared, analyzed, and interpreted to address the research questions. In this section, we present our findings according to each research question.

RQ-1: What methods are used for service discovery?

This research question explores the methods employed for web service discovery. The literature reveals a predominance of syntactic, semantic-conscious, and ontology-based exploration methodologies. Syntactic techniques, which are the earliest and rely on keyword-based strategies, are still in use. This method involves comparing keywords with web service descriptions, making the choice of keywords critical to the discovery process's success. If the keywords are insufficient or inaccurate, users may either find too many irrelevant services or none at all. The use of informal language in interrogative phrases is discouraged in this approach, as it complicates the process of linking everyday language to web service meanings.

The semantic method emerged after the syntactic approach and focuses on the semantic definition of services, allowing users to locate desired services more efficiently and accurately. This method leverages semantic web technologies like OWL-S, WS-Policy, and WSDL-S to identify web services. Instead of direct matching, the semantic-conscious method uses semantic inferences to assess the similarity between the query and service descriptions, enabling users to use natural language rather than relying on precise keywords. Additionally, this approach groups services with similar characteristics based on their descriptions.

The ontology-based approach, which evolved from semantic methods, employs ontology techniques to help users find the most relevant services in context. This method involves creating service profiles that contain detailed information about each service's functions, operations, and availability. Unlike other methods, the ontology-based approach also considers state changes, inputs, and outputs, emphasizing predictive and effect-oriented matching.

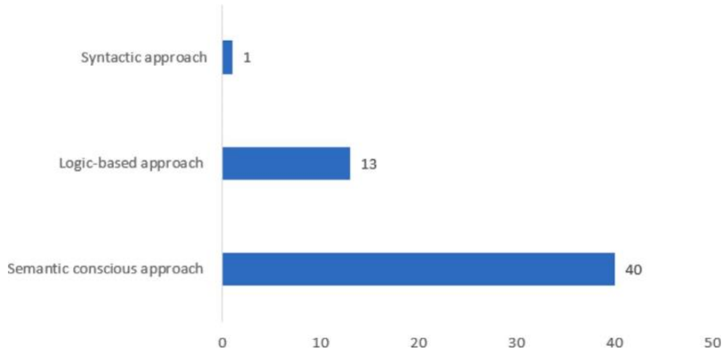

Figure 4: Distribution of approach types of publications.

When reviewing the selected articles, the semantic-conscious method was used 40 times, the ontology-based approach 13 times, and the syntactic approach only once. Figure 4 illustrates the distribution of service discovery methods utilized in the literature, highlighting that the semantic-conscious method is the most commonly employed for web service discovery. The limited representation of syntactic techniques in Figure 4, with only one publication included, likely reflects the specific criteria applied during the selection process. It is possible that the publications meeting our criteria did not extensively focus on syntactic techniques or presented their results differently.

V. CONCLUSION

As the number of available web services increases, finding useful ones becomes more challenging. Services may become inaccessible or change their functionalities over time, and a load balancer might be needed if multiple services offer the same function. Users often overlook services they cannot find. Researchers addressing these fundamental issues have guided the development of more effective discovery methods. The shift from traditional syntactic approaches, which rely on keyword matching, to more semantically-aware methods is evident. Literature on critical aspects like "Health check," "load balancing," and "service discovery" shows that service discovery remains a prominent area of research. Current studies often utilize clustering methodologies and algorithms, with Latent Dirichlet Allocation (LDA) and WordNet being popular choices. However, there is a notable gap in research concerning the application of cloud computing to service discovery, despite its rapid advancement. It is anticipated that as cloud computing evolves, its role in service discovery will become more significant. Research into online services often involves clustering and indexing, but these methods frequently yield inactive results, suggesting that researchers may prefer approaches that require user queries. For future work in web service discovery, exploring other methods not extensively covered in existing literature could be beneficial. For instance, the Semantic-Aware Web Service Discovery Approach, which uses semantic inferences, has proven effective in improving discovery accuracy. Researchers might also investigate advanced classification algorithms like Boosting and Convolutional Neural Networks (CNN), as well as techniques such as Latent Semantic Analysis (LSA) and ConceptNet, which capture semantic relationships and similarities. By incorporating these advanced approaches, researchers can enhance the effectiveness of web service discovery.

REFERENCES

1. D.H. Elsayed, A. Salah, Semantic web service discovery: a systematic survey, in: 2015 11th International Computer Engineering Conference, ICENCO, IEEE, 2015, pp. 131–136.
2. R. Phalnikar, P.A. Khutade, Survey of QoS based web service discovery, in: 2012 World Congress on Information and Communication Technologies, IEEE, 2012, pp. 657–661.
3. C. Pautasso, E. Wilde, RESTful web services: principles, patterns, emerging technologies, in: Proceedings of the 19th International Conference on World Wide Web, 2010, pp. 1359–1360.

4. W. Rong, K. Liu, A survey of context aware web service discovery: from user's perspective, in: 2010 Fifth Ieee International Symposium on Service Oriented System Engineering, IEEE, 2010, pp. 15–22.
5. V.X. Tran, H. Tsuji, A survey and analysis on semantics in QoS for web services, in: 2009 International Conference on Advanced Information Networking and Applications, IEEE, 2009, pp. 379–385.
6. Asuvaran & S. Senthilkumar, "Low delay error correction codes to correct stuck-at defects and soft errors", 2014 International Conference on Advances in Engineering and Technology (ICAET), 02-03 May 2014. doi:10.1109/icaet.2014.7105257.
7. Aziz A., Hanafi S., and Hassanien A., "Multi-Agent Artificial Immune System for Network Intrusion Detection and Classification," in Proceedings of International Joint Conference SOCO'14-CISIS'14-ICEUTE'14, Bilbao, pp. 145-154, 2014.
8. B. Kitchenham, P. Brereton, M. Turner, M. Niazi, S. Linkman, R. Pretorius, D. Budgen, The impact of limited search procedures for systematic literature reviews—A participant-observer case study, in: 2009 3rd International Sym- posium on Empirical Software Engineering and Measurement, IEEE, 2009, pp. 336–345.
9. Senthilkumar Selvaraj, "Semi-Analytical Solution for Soliton Propagation In Colloidal Suspension", International Journal of Engineering and Technology, vol, 5, no. 2, pp. 1268-1271, Apr-May 2013.
10. J. Kopecky`, T. Vitvar, C. Bournez, J. Farrell, Sawsdl: Semantic annotations for wsdl and xml schema, IEEE Internet Comput. 11 (6) (2007) 60–67.
11. A. Renuka Devi, S. Senthilkumar, L. Ramachandran, "Circularly Polarized Dualband Switched-Beam Antenna Array for GNSS" International Journal of Advanced Engineering Research and Science, vol. 2, no. 1, pp. 6-9; 2015.
12. M. Malaimalavathani, R. Gowri, A survey on semantic web service discovery, in: 2013 International Conference on Information Communication and Embedded Systems, ICICES, IEEE, 2013, pp. 222–225.
13. Aziz A., Salama M., Hassanien A., and Hanafi S., "Detectors Generation Using Genetic Algorithm for A Negative Selection Inspired Anomaly Network Intrusion Detection System," in Proceedings of Federated Conference on Ensemble Voting based Intrusion Detection Technique using Negative Selection Algorithm 157 Computer Science and Information Systems, Wroclaw, pp. 597-602, 2012.
14. Catal, On the application of genetic algorithms for test case prioritization: a systematic literature review, in: Proceedings of the 2nd International Workshop on Evidential Assessment of Software Technologies, 2012, pp. 9–14.