

Meet User's Service Requirements in Smart Cities Using Recurrent Neural Networks and Optimization Algorithm

Seyed Salar Sefati¹, Graduate Student Member, IEEE, Bahman Arasteh, Simona Halunga, Member, IEEE, Octavian Fratu, and Asgarali Bouyer²

Abstract—Despite significant advancements in Internet of Things (IoT)-based smart cities, service discovery, and composition continue to pose challenges. Current methodologies face limitations in optimizing Quality of Service (QoS) in diverse network conditions, thus creating a critical research gap. This study presents an original and innovative solution to this issue by introducing a novel three-layered recurrent neural network (RNN) algorithm. Aimed at optimizing QoS in the context of IoT service discovery, our method incorporates user requirements into its evaluation matrix. It also integrates long short-term memory (LSTM) networks and a unique black widow optimization (BWO) algorithm, collectively facilitating the selection and composition of optimal services for specific tasks. This approach allows the RNN algorithm to identify the top- K services based on QoS under varying network conditions. Our methodology's novelty lies in implementing LSTM in the hidden layer and employing backpropagation through time (BPTT) for parameter updates, which enables the RNN to capture temporal patterns and intricate relationships between devices and services. Further, we use the BWO algorithm, which simulates the behavior of black widow spiders, to find the optimal combination of services to meet system requirements. This algorithm factors in both the attractive and repulsive forces between services to isolate the best candidate solutions. In comparison with existing methods, our approach shows superior performance in terms of latency, availability, and reliability. Thus, it provides an efficient and effective solution for service discovery and composition in IoT-based smart cities, bridging a significant gap in current research.

Index Terms—Black widow optimization (BWO) algorithm (BWO), Internet of Things (IoT), recurrent neural network (RNN), service discovery, smart cities.

Manuscript received 10 April 2023; revised 30 June 2023 and 24 July 2023; accepted 1 August 2023. Date of publication 8 August 2023; date of current version 7 December 2023. This work was supported by the Project “Mobility and Training for Beyond 5G Ecosystems (MOTOR5G)” which received funding from the European Union's Horizon 2020 Programme under the Marie Skłodowska Curie Actions (MSCA) Innovative Training Network (ITN) under Grant 861219. (Corresponding author: Seyed Salar Sefati.)

Seyed Salar Sefati, Simona Halunga, and Octavian Fratu are with the Telecommunications Department, Faculty of Electronics, Telecommunications and Information Technology, University POLITEHNICA of Bucharest, 060042 Bucharest, Romania (e-mail: Sefati.seyedsalar@upb.ro; Simona.halunga@upb.ro; Octavian.Fratu@upb.ro).

Bahman Arasteh is with the Department of Software Engineering, Faculty of Engineering and Natural Science, Istinye University, 34010 Istanbul, Türkiye (e-mail: bahman.arasteh@istinye.edu.tr).

Asgarali Bouyer is with the Department of Software Engineering, Faculty of Computer Engineering, Azarbaijan Shahid Madani University, Tabriz 5375171379, Iran, and also with the Department of Software Engineering, Faculty of Engineering and Natural Science, Istinye University, 34010 Istanbul, Türkiye (e-mail: a.bouyer@azaruniv.ac.ir).

Digital Object Identifier 10.1109/JIOT.2023.3303188

I. INTRODUCTION

SMART cities are becoming popular as cities around the world want to improve their infrastructure and services [1]. The Internet of Things (IoT) combines different technologies and is essential for making smart cities work [2]. IoT-based smart cities leverage sensors, connectivity, and automation to optimize infrastructure and resource consumption. By gathering data from connected devices and sensors in the city, IoT technology is used to improve sustainability, optimize energy consumption, and enhance residents' quality of life [3]. In addition, the IoT is vital in connecting various devices within intelligent wearable devices and agriculture. While traditional mobile networks have mainly been developed for human communication, IoT networks aim to link multiple objects without human intervention [4]. Examples of IoT network applications include control systems, intelligent recognition, positioning, and monitoring.

In IoT, service discovery and composition involve connecting different IoT devices, which requires using various technologies, such as the semantic Web, service-oriented architecture, and cloud computing [5]. These technologies enable the development of a flexible and adaptable service infrastructure, that can meet the changing requirements of the smart city [6]. Service discovery and composition can be used in traffic management systems [7], air pollution evaluation [8], and environmental monitoring applications [9]. One of the main challenges of service discovery and composition in IoT is to ensure that the services selected meet the user's requirements in terms of Quality of Service (QoS) [10].

The main goal of this research is to present a unique method for selecting the services that will best satisfy the consumers' QoS expectations. Discovering IoT smart cities services based on existing circumstances with human input is an NP-hard issue [11]. The user's needs significantly impact the service discovery, and QoS values, and that impact vary for each service. The recurrent neural network (RNN) [12] algorithm estimates the QoS of each service based on the number of connected devices, traffic load, network topology, using historical data and learned patterns. Black widow optimization (BWO) [13] selects the best services according to the fitness function of spiders and compounds these services. In Section IV, the proposed method is discussed deeply. Briefly, the current article's contributions are as follows.

- 1) To achieve the most suitable integration of QoS, that fits best with the users' needs.
- 2) To optimize the latency, energy consumption, availability, and reliability.
- 3) To perform a series of analyses to assess the functionality of the method across different services.

The structure of this article is organized as follows. Section II offers a review of the relevant recent literature. In Section IV, the problem statement is presented, and the proposed method is discussed in detail. The simulation setup and the results are presented in Section IV, and in section IV several interesting conclusions are highlighted, as well as recommendations for future research.

II. RELATED WORKS

Service discovery and composition are critical challenges in the field of IoT, as they enable the efficient and effective use of the diverse set of services and devices available in IoT environment. This study divided the related work into deterministic and nondeterministic algorithms.

Deterministic algorithms are the ones that suggest service discovery and composition algorithms for IoT that guarantee a deterministic outcome. Such algorithms use well-defined rules and procedures to match services with consumers' requirements, ensuring the same output every time the algorithm is run [14].

Hayyolalam et al. [15] proposed three phases for developing such algorithms, namely, service description, service composition, and service execution. The framework of the system is selecting the best candidate services that match the user's QoS and Quality of Experience (QoE) requirements in the first phase. The second phase combines the selected services into a composite service that meets the user's requirements while considering the resource constraints and dependencies between services. Finally, the composite service is executed and monitored at the desired QoS and QoE levels. The advantage of this approach is its reliability, availability, and user satisfaction by considering QoS and QoE; however, it is suffering from high-programming complexity and low number of users.

Deng et al. [16] introduced a framework for service discovery utilizing the sentence bidirectional encoder representations from transformers (SBERT) model and the graph attention network (GAT) algorithm. By embedding service descriptions into a high-dimensional vector space, the SBERT model enables efficient comparison and similarity evaluations between services. The GAT algorithm aids in constructing a service graph based on the semantic resemblance between service descriptions, thereby improving service discovery and selection efficiency. While this method facilitates self-organizing collaboration among diverse IoT devices without the need for a computing infrastructure, enhancing both security and scalability, it suffers from lengthy execution times and increased latency.

Mohana et al. [17] proposed a method for discovering services in a Social Internet of Things (SIoT) environment using semantic rules. SIoT refers to the integration of IoT with social networks, where people can interact with smart

devices and services through social networks. The proposed method consists of two main components: 1) a service discovery engine (SDE) and 2) a semantic rule engine (SRE). The SDE is responsible for discovering available services in the SIoT environment, while the SRE applies semantic rules to filter and prioritize the discovered services based on user preferences and requirements. The SRE uses ontologies to represent user preferences and requirements. This technique provides good efficiency with respect to personalization and scalability, but may face challenges related to complexity, semantic interoperability, and resource requirements.

Mezni et al. [18] proposed a framework for smart monitoring of water environments using a combination of sensors, cloud computing, and service-oriented architecture. The proposed framework consists of three main layers: 1) the physical layer; 2) the network layer; and 3) the application layer. This article combines knowledge graph technology and network representation learning (NRL) for water quality control and drinking water consumption optimization by using intelligent corrective measures and management strategies. The knowledge graph-like water information network (WIN) is mapped incrementally into a lower dimensional vector space that is constantly readjusted to take into account all changes or problems that appeared in monitored water zones, which facilitates decision-making on appropriate management actions. This technique provides real-time monitoring and high scalability but may face high cost, complexity, as well as issues related to data privacy and security challenges.

Urbietta et al. [19] suggested a service composition framework that uses an abstract service model called wEASEL, which allows for dynamic reasoning on user tasks and service behavior. Its main objective is to enable mobile users to integrate IoT-based services available in their vicinity. The framework is evaluated using a new testbed based on OWLS-TC4. In the service selection step, the method selects the most appropriate services based on the user requests and on the services available in the smart city environment. In the second step, the selected services create a composite service that fulfills the user request. The proposed method has the potential to provide personalized and relevant services to users but does not address potential privacy or security concerns with the collection and processing of user context data.

Nondeterministic algorithms are the ones that allow multiple outcomes or paths to a solution rather than a single deterministic outcome. Nondeterministic algorithms are used in specific fields of computer science, such as computational complexity theory and cryptography. In addition, these algorithms are often used for optimization problems, where the goal is to find the best or optimal solution [20].

Bidi et al. [21] proposed a method, called GA-based QoE-aware service composition (GQSC), that is used in solutions for service composition in smart city environment. In this article, the authors use a fog computing platform that provides resources and services for smart cities. In addition, the fog computing platform is designed to be fault-tolerant, which means that it can continue to provide services even in the event of a failure or outage. The proposed method also considers the QoE of the services, measured based on user satisfaction. The

proposed method provide high reliability and high accuracy, but may face challenges related to high-energy consumption and high latency.

Hosseinzadeh et al. [22] proposed a method called hybrid artificial neural network-based particle swarm optimization (ANN-PSO) for the service composition issue. In this article the method proposed uses a decision-making process that considers both the QoS requirements and the available resources in the cloud-edge computing environment. Furthermore, it considers the context of the IoT devices and services in the cloud-edge computing environment, which can help providing personalized and relevant services to users. This technique has a high availability, reliability, and low cost but suffers from high-energy consumption and high complexity in programming.

Neiat et al. [23] proposed an IoT service composition technique that uses deep reinforcement learning (DRL) approach, that learns from user experience to improve the service composition process. The authors consider the mobility of IoT devices and the QoS requirements of the services in each process. This approach also uses dynamic programming to optimize the service composition process and to handle the IoT environment's uncertainty and randomness. This technique proves to have high scalability and accuracy but it suffers from long execution time and high-convergence time.

Namoun et al. [24] proposed a two-phase machine learning (ML) algorithm that considers the context of the user and the QoS requirements of the selected services. The first phase of the framework involves training a ML model on historical data to predict the QoS of the services. The second phase involves using the predicted QoS values to select the most appropriate services for the user. The proposed method also considers the user's context, including their disabilities, preferences, and location. It is shown that this technique has a high scalability and accuracy, but it is suffering from low latency and low reliability.

Sefati and Navimipour [25] proposed a service composition technique that uses a hidden Markov model (HMM) and ant colony optimization (ACO) to improve QoS. First, the HMM model is trained to anticipate QoS, while the Viterbi algorithm improves the emission and transition matrices. Then, the ACO algorithm is employed to assess the QoS and to select the most appropriate route. The method proposed has a good efficiency in availability, response time, cost and reliability but it needs high energy use and high complexity for the HMM algorithm.

Table I shows the advantages and disadvantage of the methods proposed in this paper discussed above.

A. Comparative Outlook

An examination of related work reveals the extensive use of deterministic and nondeterministic algorithms in the realm of IoT service discovery and composition. Deterministic algorithms have proven their worth in terms of reliability and user satisfaction, despite challenges related to programming complexity and user capacity. Conversely, nondeterministic algorithms offer excellent optimization capabilities but are marked by high-energy consumption, extended execution time, and complexity. A closer look reveals a gap in the current

TABLE I
PROS AND CONS OF RELATED WORKS

Authors	Proposed method	Advantage	Disadvantage
Hayyolalam, et al. [15]	Three-phase deterministic algorithm	High reliability, availability, and user satisfaction	High programming complexity Low number of users
Deng, et al. [16]	SBERT & GAT	Highly secure and high scalable	High execution time and high latency
SD, et al. [17]	Semantic rule-based method	High efficiency in personalization and high scalability	High complexity, and resource requirements
Mezni, et al. [18]	Graph technology and NRL	Real-time monitoring and high scalability	High cost, high complexity, low data privacy and security
Urbietta, et al. [19]	wEASEL	Personalized and provide relevant services to users	Does not address potential privacy or security concerns with the collection and processing of user context data
Hosseini Bidi, et al. [21]	GQSC	Provides reliable and high-quality services to users	High energy consumption and high latency
Hosseinzadeh, et al. [22]	ANN-PSO	High availability, reliability and low cost	High energy consumption High programming complexity
Neiat, et al. [23]	DRL approach	High scalability and High accuracy	Long execution time and high convergence time
Namoun, et al. [24]	ML algorithm	High scalability and high accuracy	Low latency and Low reliability.
Sefati and Navimipour [25]	HMM and ACO algorithm	High availability, reliability, Low response time, Low cost	High energy consumption High complexity

literature, with few studies considering the dynamic QoS expectations of consumers across diverse services. Crucial issues, such as latency, energy consumption, and availability, are either not fully addressed or are tackled at the expense of other significant parameters. The research presented here bridges this gap by proposing an innovative approach that fuses RNN and black widow optimization (BWO) for efficient service discovery in smart cities' IoT systems. Our method considers the dynamic QoS expectations of consumers and optimizes service selection accordingly. Emphasizing reducing latency and energy consumption, it also enhances availability and reliability. Moreover, the presented research evaluates the functionality of our method across various services, adding to its robustness. This unique aspect allows for an optimized and comprehensive approach to service discovery and composition in smart city environments. In doing so, it considers consumers' dynamic QoS expectations and the complex tradeoffs between different performance metrics.

III. PROPOSED METHOD

The technique proposed in this work aims to anticipate the QoS for different services. It uses RNN and BWO algorithms to provide users with the best achievable services. RNN is a

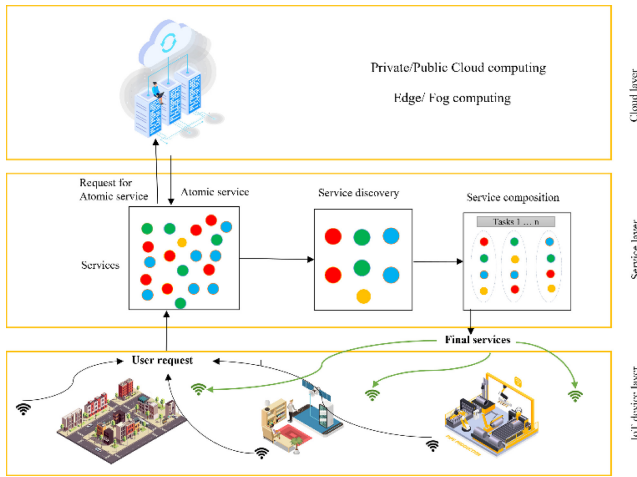


Fig. 1. Service discovering and composition in smart city.

type of neural network appropriate for analyzing sequential data and can be utilized to predict QoS values based on historical data [26]. The BWO algorithm is a swarm intelligence algorithm inspired by the behavior of black widow spiders that can be used to optimize the selection and composition of services based on user needs [27]. One potential advantage of this approach is that it could help reduce the time and effort required for users. However, it is essential to note that the effectiveness of this approach will depend on various factors, such as the quality of the data, used for training the RNN and the accuracy of the QoS predictions, made by the model.

A. Problem Statement

The problem statement of service discovery and composition involves the challenge of identifying and integrating services from various sources in a smart city environment. These issues include heterogeneity of service interfaces, lack of standardization, scalability, security, and automated methods. Solutions for this problem require the development of innovative technologies that can support the seamless integration and orchestration of services across different domains. Service discovery requires a mechanism to find services provided by IoT devices and assess their availability and capabilities. This can be a complex issue since the IoT devices may have limited computing resources and connectivity. Service composition in IoT requires a flexible approach to handle many devices and services involved and to ensure real-time collaboration. As shown in Fig. 1, the IoT device layer is connected to the service layer, and, with the help of the RNN algorithm, we discover the QoS of services; then, with the use of the BOW algorithm, we composite these services. An atomic service in IoT discovery and composition refers to a basic, indivisible service that provides a specific function or capability. These services can be combined or orchestrated to create more complex and valuable services for IoT applications [28].

QoS-aware service composition and discovery can fulfill the customer requirements through a quadruple of (WF, QP, QC, and TC). WF stands for workflow, which represents the

practical requirements with multiple tasks. QP stands for QoS performance, representing the users' priorities for each QoS factor. QC stands for QoS constraints, which represent the universal QoS constraints for each factor. Finally, TC stands for transactional constraints, and represents the users' need for the transactional conduct of the composite service.

The composition engine searches for candidate services for each job in the workflow by matching the functional requirements with the service details. The main goal is to create a composite service that satisfies users' end-to-end QoS and transactional constraints. For each task, a set of candidate services is identified and bound to create the composite service.

- 1) The practical requirements of the users are denoted by WF, which is composed of multiple tasks that are interconnected through composition structures.
- 2) QP is a set (P_1, \dots, P_d) that reflects the user's priorities for each QoS factor.
- 3) QC is a collection (c_1, \dots, c_d) that represents the universal QoS constraints of the users for each QoS parameter. It defines an upper limit constraint for negative factors and a lower limit constraint for positive factors. If a constraint, c_d , is null, it indicates that there is no constraint for the corresponding.
- 4) TC reflects the user's need for the transactional conduct of the composite service. The final operator can recognize the transactional characteristic of the composite service, as well as the user need not specify this constraint. A composite service can combine the existing services in order to provide a new, value-added composite service.

B. QoS of Service Discovery and Composition

Integrating QoS into service composition and discovery is an essential aspect of service oriented computing (SOC). A significant challenge in achieving this goal is selecting the most suitable combination of services that meet the user's defined global QoS requirements. This challenge becomes even more pronounced in dynamic service environments, where two additional problems arise.

Cost: The cost in service discovery and composition refers to the overall expense or effort required to discover, select, and compose different services in order to fulfill a specific task or goal. Equations (1) and (2) demonstrate the overall cost of a service

$$C_{\text{service}}(S_i) = C_{\text{sp}}(S_i) + \sum_{m=1}^{D(S_{i,m})} [C_{\text{dp}}(D_{i,m}) + C_{\text{LD}}(D_{i,m})] \quad (1)$$

where $C_{\text{sp}}(S_i)$ represents the cost of selecting the service provider (SP) for the service S_i , $D(S_{i,m})$ is the set of data required by service S_i , $C_{\text{dp}}(D_{i,m})$ represents the cost of accessing the required, and $C_{\text{LD}}(D_{i,m})$ represents the total cost of transferring over the network

$$C_{\text{total}}(\text{CS}) = \sum_{i \in P} C_{\text{service}}(S_i) + \sum_{k \in C_L} C_L(l, k). \quad (2)$$

Equation (2) determine the total cost that is calculated as the sum of the cost of each service in the set $\{P\}$, denoted by $C_{\text{service}}(S_i)$, and the cost of each link in the set C_L , denoted by $C_L(l, k)$ [29].

Reliability: In IoT systems reliability is a critical factor for ensuring the efficient operation of connected devices and the delivery of valuable services to end-users. Equation (3) determines the reliability of the IoT services

$$R(t) = \frac{1}{(a-1)}(\lambda t)^{(a-1)}e^{(-\lambda t)} \quad (3)$$

where $R(t)$ represents the probability density function (PDF) of the time t that it takes for a system to fail (i.e., the time between service discovery and system failure), and a controls the degree of skewness or asymmetry of the PDF. A larger value of a indicates a more symmetric distribution, while a smaller value of a leads to a more skewed distribution. The λ parameter is the failure probability at t time [4].

Equation (4) determines the expected value, $E(t)$ that represents the average time until failure, also known as the mean time to failure (MTF). Here, t represents the time until failure, $f(t)$ is taken over the entire range of possible failure times

$$E(t) = \int_0^{\infty} tf(t)dt. \quad (4)$$

By calculating the expected value we can estimate the average time until failure, which can be used to make decisions regarding maintenance, repair, and replacement of IoT devices and systems.

Availability: In the context of service discovery in IoT, the availability represents the proportion of time in which a service or system is operational and accessible. Equations (5) and (6) evaluates the availability, as follows:

$$A(t) = \frac{\mu}{\lambda + \mu} + \frac{\mu}{\lambda + \mu}e^{-(\lambda + \mu)t} \quad (5)$$

$$\lim_{t \rightarrow \infty} A(t) = \frac{\mu}{\lambda + \mu} = \frac{\text{MTF}}{\text{MTF} + \text{MTR}} \quad (6)$$

where λ represents the failure rate and μ repair rate. As the operating time t trends to infinity the availability $A(t)$ trends to the probability of availability. These formulas can be used to determine the MTF and mean time to repair (MTR) [4].

Latency: In IoT systems, the latency refers to the time necessary for a device or an application to discover available services or resources on the network. Equation (7) determines the latency of IoT as follows:

$$\tau_1 = \tau_t + \tau_{\text{prop}} + \tau_{\text{proc}} + \tau_{\text{ret}} + \tau_{\text{sig}} \quad (7)$$

where τ_1 shows the total end-to-end latency, where τ_t is the time necessary to transmit the data over the network, τ_{prop} is the propagation delay (i.e., the time it takes for the data to get from the transmitter to the receiver), τ_{proc} is the processing delay time, which is the time it takes for the receiver to process the data once it has been received, τ_{ret} is the queuing delay to be processed, and τ_{sig} is the time necessary to send and receive control signals for synchronization message and error control checking [30].

Energy Consumption: IoT service discovery involves searching for available services, which can be energy intensive process, since the devices are frequently sending and receiving messages to maintain an up-to-date service directory. Equation (8) determines the energy usage of IoT-capable devices

$$E = \frac{E_{\text{init}} - E_c}{E_{\text{init}}} \quad (8)$$

where E_{init} is representing the initial energy of each node and E_c is representing the current energy of each node [30].

C. Standardization of Qualitative Attributes

Standardization of qualitative attributes refers to the way in which nonnumeric characteristics of a product, service, or process can be measured and defined consistently and objectively. This can include attributes, such as usability, reliability, and user satisfaction, which are essential for evaluating the quality of a product or service. There are several standardization organizations that have developed guidelines and frameworks for measuring and evaluating qualitative attributes. Overall, standardization of qualitative attributes can help ensure consistency and reliability in evaluating product and service quality, which can be beneficial for both producers and consumers. A solution for standardization was proposed in [31]

$$q_{si}^k = \begin{cases} \frac{q_{si}^k - \text{Min}(q_{si}^k)}{\text{Max}(q_{si}^k) - \text{Min}(q_{si}^k)}, & \text{for positive attributes} \\ \frac{\text{Max}(q_{si}^k) - q_{si}^k}{\text{Max}(q_{si}^k) - \text{Min}(q_{si}^k)}, & \text{for negative attributes} \end{cases} \quad (9)$$

where q_{si}^k shows to the k_{th} quality of the i_{th} service, while $\text{Max}(q_{si}^k)$ and $\text{Min}(q_{si}^k)$ are the high values of the k_{th} for the i_{th} service [31].

D. Proposed Method

It is computationally challenging to discover IoT intelligent city services that match current circumstances and incorporate human input. The user's needs significantly impact service discovery, and QoS values vary for each service. The RNN algorithm estimates [32] the QoS of each service based on the number of connected devices, traffic load, and network topology, using historical data to predict QoS based on learned patterns [33]. The BWO algorithm can be applied to service composition in the second step. The proposed method will be described in more detail in the following section.

In the RNN algorithm, there are three levels (input layer, recurrent layer, and output layer). The input layer for the method proposed in this work consists of the users preferences, number of users, number of services, target services, and top- K services. The RNN algorithm could be used to find the top- K services by processing data based on different services QoS and under different network conditions. The RNN could make predictions with respect to different services QoS and determine the top- K services to be used for a particular task. Top- K can be helpful in a variety of IoT scenarios, such as in resource-constrained environment, where it is essential to minimize the overhead of service discovery, or in situations where there is a large number of possible services to choose from,

and it is impractical to evaluate all of them. Equation (10) determines the RNN input as the formula in this proposed method

$$x_s = f_n(w_n(f_{n-1}(\dots f_1(w_1 R_S^T + b_1) \dots) + b_{n-1}) + b_n). \quad (10)$$

A particular set of RNN layers uses nonlinear learning to acquire the concept of x_s . Here, R_S^T represents a row matrix containing the QoS values between the top- K source services (S) and target services (T) in the context of IoT service discovery. The top- K services refer to the highest ranked services based on specific QoS criteria. The QoS values provide a method to evaluate and compare different IoT services, essential for service discovery, ranking, and selection. The elements of the R_S^T matrix are the individual QoS values between each top- K source service and target service pair. These values can include various performance metrics, such as response time, throughput, reliability, availability, and energy consumption. The matrix aims to provide a more efficient and manageable representation of the most relevant services for the service discovery task by focusing on the top- K services. The importance of the R_S^T matrix lies in its role as input data for the RNN for discovering and ranking IoT services. By processing the QoS values in the R_S^T matrix, the model can learn complex patterns, relationships, and dependencies between the top- K services and the target services. The n th layer of the model uses the variables w_n , b_n , and f_n . Here, w_n is the weight of service discovery in IoT, used to adjust the influence of various inputs on the prediction of the QoS provided by different services. By adjusting the weights, the RNN can learn the relationships between the inputs and the QoS provided by different services and make more accurate predictions about the best service to use for a particular task. b_n denotes the bias, which is a variable in charge to deal with the systematic errors in the network conditions data. By adjusting the biases, the RNN can improve the overall accuracy of predictions about the best service. We denote by f_n the activation function in an RNN for service discovery in IoT. This function can have a significant impact on the performance of the network and determines how inputs are transformed into outputs.

The hidden layer long short-term memory (LSTM) is trained give as answer the solution according to the input function. During training, the backpropagation through time (BPPT) technique is used to update the RNN parameters. Equations (11) and (12) shows how the first training in hidden states is performed

$$h_t = \sigma(W \cdot x_t + U \cdot h_{t-1} + b) \quad (11)$$

$$y_t = W \cdot h_t \quad (12)$$

where h_t represents the hidden state at time step t , namely, the internal state of the RNN, showing the relevant temporal patterns and relationships between devices and services up to the current time step. σ shows the nonlinear activation function, usually a sigmoid or hyperbolic tangent (tanh) function. It introduces nonlinearity into the model, allowing it to learn complex patterns in the data. W shows the input of hidden weight matrix, that contains the weights that control how the input data at each time step x_t influences the hidden state h_t . In IoT service discovery, these weights assist in

determining the importance of input features, such as device attributes or sensor readings, in predicting device behavior or service requirements. U is the hidden-to-hidden weight matrix, that contains the weights that control the way in which the hidden state from the previous time step h_{t-1} influences the current hidden state h_t . In IoT service discovery, these weights help the RNN model capture the temporal relationships and dependencies between devices and services. h_{t-1} shows the hidden state at the previous time step ($t-1$), and it represents the internal state of the RNN at the previous time step, encoding the temporal patterns and relationships between devices and services up to that point.

Equation (11) shows the output of the node at time step t , where y_t is the output, and multiplying the weight matrix W with the final hidden state h_t . The product of the weight matrix W and the hidden state h_t generates the output at time step t . Equation (12) shows that the length of a data sequence (number of services, number of users, and top- K services) significantly impacts how long an RNN takes to train.

RNNs' ability to remember previous states makes them suitable for IoT service discovery, allowing them to consider previous inputs and make informed decisions about service selection. The forget gate in RNNs can control the influence of previous selections on the current decision. The forget gate formula, given in (13), determines how previous selections should be considered or ignored

$$f_t = \sigma(W_f[h_{t-1}x_t] + b_f) \quad (13)$$

where f_t represents the forget gate activation, σ is the activation function, and W_f is the forget gate weight matrix. In IoT service discovery, W_f elements are specific to the problem and learned during RNN training. The matrix is initialized randomly and updated iteratively through backpropagation to minimize error. Values depend on input data and RNN architecture, optimized during training for improved prediction accuracy. In the context of IoT, the forget gate weight matrix can be used to help the RNN model to predict time series data. This can help the RNN to learn long-term dependencies and make accurate predictions over longer time horizons. $h_{t-1}x_t$ represents the chain of the previous hidden state h_{t-1} and the current input x_t , and b_f is the forget gate bias.

To minimize the impact of bias in RNNs, it is important to carefully consider the training data used to develop the network and to use representative and unbiased data [34]. To verify that the network is producing accurate predictions, tracking the network's performance over time and modifying weights is crucial

$$i_t = \sigma(W_i[h_{t-1}x_t] + b_i) \quad (14)$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}x_t] + b_c) \quad (15)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t. \quad (16)$$

In (14) i_t is a input gate activation and calculating the sigmoid function σ , which determines how much of the new input information x_t should be let into the cell state c_t at time step t . The variable i_t ranges from 0 to 1, where 0 means that no information from the current input x_t is let into the new cell state c_t , and 1 means that all of the information from

x_t is let in. In the context of IoT and service discovery, this can be used to incorporate new information obtained from IoT sensors into the RNN model to make predictions about future events or states. W_i is typically used to represent the weight matrix of maps concatenation of the previous hidden state and the current input to the current hidden state h_{t-1} . For example, considering an IoT application that uses sensor readings to predict air quality levels in a city. The concatenation of the previous hidden state and the current input could include information about the air quality levels at previous time steps and other factors that could affect air quality, such as temperature, humidity, and wind speed. The weight matrix W_i could then be used to map this concatenated input to the current hidden state, allowing the RNN to learn patterns in the data and make predictions about future air quality levels. b_i represents the bias term for the input gate. The bias term allows the network to learn a shift in the activation function, which can be useful for improving the model's predictions. The value of b_i is learned during training and can be modified through backpropagation to minimize the error between the predicted output and the actual output. In the context of IoT service discovery, the value of b_i would be specific to the problem being addressed and would be optimized during the training process to improve the accuracy of the model's predictions. W_c represents the weight matrix of candidate hidden state. The candidate hidden state represents the updated state of the system based on the current input and the previous hidden state. The tanh activation function maps its input to the range of $[-1, 1]$, which makes it particularly well suited for use in RNNs. b_c is a learnable parameter that helps adjust the candidate's hidden state and improve the accuracy of the model's predictions. It helps to shift the activation function of the tanh nonlinearity to better fit the input data, leading to better performance of the RNN model. \tilde{C}_t is the updated value of the memory cell at time step t

$$h_t = o_t \cdot \tanh(c_t). \quad (17)$$

The current hidden state h_t is computed by multiplying o_t and the hyperbolic tangent of c_t . The output gate activation o_t is a parameter in the LSTM network used for processing sequence data.

Normalization is a typical preprocessing step RNNs for IoT service discovery. Normalization helps to standardize the input data and to ensure that the input values are within a consistent range [35]. This can help to prevent numerical instability and improve the performance and accuracy of the RNN. LSTM is being used to update the hidden state of the RNN at each time step t . The LSTM processes is used input gate, forget gate, and output gate, which allow it to selectively remember or forget information from previous time steps and update the hidden state based on the current input [36]. Normalization is typically performed on the input data before it is fed into the RNN and can significantly impact the network's performance and accuracy. By normalizing the input data, it is possible to improve the robustness and stability of the network and to ensure that the network is making accurate predictions for IoT

service discovery

$$h_t^{(i)} = \text{LSTM}\left(x_t^{i-1} h_{(t-1)}^{(i)}\right) \quad \text{where } t = 1, 2, \dots, n \quad (18)$$

where x_t^{i-1} refers to the input data for time step t in the $(i-1)$ th iteration.

LN is a technique that can be used to normalize the inputs to each layer of a neural network, which can help to improve the training and performance of the network. The function of LN is to normalize the inputs to each layer of the RNN by subtracting the mean and dividing by the standard deviation of the inputs. This can help to reduce the impact of differences in scale and distribution between the input features, which can improve the training and performance of the RNN, where x_t^i represents the output of the normalization layer at time step t in the i th iteration, h_t^i represents to the hidden state of the RNN at time step t in the i th iteration. At the end input $h_{t1}^{(i)}$ returns the normalized output x_t^i

$$x_t^{(i)} = \text{LN}\left(h_{t1}^{(i)}\right) \quad \text{where } t = 1, 2, \dots, n. \quad (19)$$

The following equations describe the normalization at the t time instant:

$$\text{LN}(h_t) = f\left[\frac{g}{\sigma_t} \Theta(h_t - \mu_t) + b\right] \quad (20)$$

$$\mu_t = \frac{1}{H} \sum_{j=1}^H h_{t,j} \quad (21)$$

$$\sigma_t = \frac{1}{H} \sum_{j=1}^H (h_{t,j} - \mu_t)^2. \quad (22)$$

The above expressions represent mathematically the normalization layer applied to the hidden state of an RNN, with the normalization function being a combination of three functions f , g , and the activation function Θ . The function f scales the hidden state h_t . g is a trainable scale parameter, and μ_t is the mean of the hidden state h_t . The activation function Θ is applied elementwise to $h_t - \mu_t$, and b is a trainable bias term. H demonstrates the hidden components in LSTM. The hidden state is a vector that represents the output of the LSTM at a particular time step, while the cell state is a vector that represents the internal memory of the LSTM. Fig. 2 shows the normalization steps in RNN.

After discovering the services, the BWO algorithm combines this service for answering the user's demand. In the context of IoT service composition, BWO can be used to find the optimal combination of services that meet the requirements of a given system. The algorithm simulates the behavior of a black widow spider in selecting its prey by exploiting the attractive and repulsive forces between different services. The attractive forces represent the benefits of selecting a particular service, while the repulsive forces represent the costs or limitations of that service [37]. The BWO algorithm starts with a population of candidate solutions, each representing a different combination of services. The algorithm then evaluates the fitness of each candidate solution using a fitness function that takes into account the attractive and repulsive forces. The best candidate solutions are then selected as "spiders" and used to

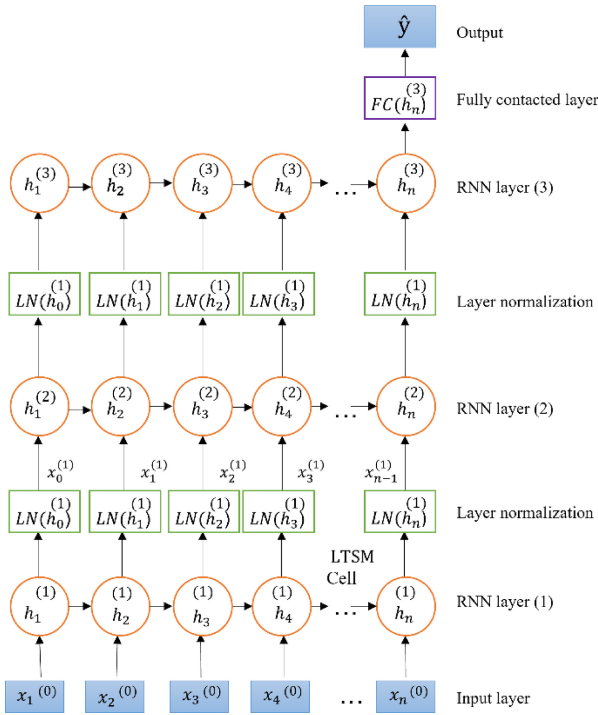


Fig. 2. Architecture of deep-RNN.

explore the search space. Over time, the algorithm iteratively improves the candidate solutions until an optimal combination of services is found.

Equation (24) shows the initial solution of the BWO population. The j th request for the service composition and i th “spider” might have the following random (rand) initialization function:

$$\text{BWO}(i, j) = SS_i[(S_{\text{ubi}} - S_{\text{lbi}}) \cdot \text{rand}() + S_{\text{lbi}}] \quad (23)$$

where S_{lbi} and S_{ubi} are the initial and final available services. SS_i determines the available service in the set of services for the user’s demands. $\text{rand}()$ is a random number generated within the range $[0,1]$. In the context of service discovery in IoT, SS_i represents the set of available services that meet the user’s demands for service i . This set can be determined mathematically based on the user’s requirements and the QoS metrics of the available services. For example, if a user requires a service that provides a response time of less than 1 s, and there are three available services with response times of 0.5, 1.2, and 1.5 s, then SS_i would be the set containing the first service that meets the user’s response time requirement.

The mathematical definition of SS_i could be expressed as follows: let $S = \{S_1, S_2, \dots, S_n\}$ be the set of available services, and let $\text{QoS}_{S_j} = \{q_1, q_2, \dots, q_m\}$ be the vector of QoS metrics for Service $_{S_j}$. The user’s demands for service i can be specified as a vector $d_i = \{d_{i1}, d_{i2}, \dots, d_{im}\}$ of desired QoS metrics for service i . Then SS_i , the set of available services that meet the user’s demands for service i , can be defined as

$$(SS)_i = \{S_j | \text{QoS}_{S_j} \text{ satisfies } d_i\}. \quad (24)$$

We consider both positive and negative QoS requirements and each spider is assigned a place for adjusting value based on the service level agreement (SLA). Additionally, the user’s priority has to be indicated in the fitness function. Equations (25) and (26) show the positive and negative QoS criteria, respectively. These QoS must be used to normalize as follows:

$$c_i^-(\text{BWO}) = \frac{Sq_i^- - q_i^-(\text{BWO})}{Sq_i^+} \quad (25)$$

$$c_i^+(\text{BWO}) = \frac{q_i^+(\text{BWO}) - Sq_i^+}{Sq_i^+} \quad (26)$$

where c_i^- and c_i^+ represents the QoS of i th services and Sq_i^- , Sq_i^+ are QoS limitations determined in the SLA. Positive QoS criteria should be increased while negative QoS criteria should be decreased in order to prevent SLA. In calculating the fitness values, we also need to take into account the user’s preferences.

Parameter ∂_i represent the sensitivity of the cost function concerning the predicted QoS metrics and is used to update the predicted QoS metrics to improve the overall performance of the system. c_i represent the learned weights in the BWO algorithm, which are used to extract meaningful features from the input data. $f(\text{BWO})$ is the fitness function of services and select the optimal combination that meets the user’s QoS requirements, where o is the quantity of QoS requirements. The suggested algorithm aims to find composite services with a high-fitness value. Equations (27) and (28) represent the fitness function as follows:

$$f(\text{BWO}) = \sum_{i=1}^o \partial_i \times c_i(\text{BWO}) \quad (27)$$

$$\sum_{i=1}^o \partial_i = 1. \quad (28)$$

The positions of the “spider” and the considering food are essential for the best composition answers.

Algorithm 1 shows the service discovery composition using the RNN and BWO algorithm.

IV. EXPERIMENTAL RESULTS

The proposed method is compared to various algorithms, such as GQSC [21], ANN-PSO [22], DRL [23], and ML [24] algorithm, which is mainly used for service discovery and composition. All algorithms showed identical fitness values for all data sets. The study examined the variables and their amounts, as well as the simulation factors and outcomes.

A. Characteristics of the Data Set

Our urban context-based approach was evaluated using the Cooja simulator for mobile applications. We used a simulated smart city environment where service providers and citizens interacted with a network of gateways that managed services. Providers, citizens, and gateways could be either static or mobile, and the gateways could perform service registration, distribution, and discovery. We implemented the overlay maintenance process for both models to adapt them to the large and

Algorithm 1 Pseudo Code of Service Discovery: A Fusion of RNN and BWO Algorithm

Input: Historical service data, Users need QoS, Number of services, Number of users (Eq. (10));

Output: Discover the highest QoS

Set

LSTM, activation, normalization, regularization, and dropout layers parameters to define the LSTM model (LM) **Model** \leftarrow LM

Initialize

RNN layer with input size, learning layer, epochs, historical service data

For epochs=1 to n **do**

While

Pass the current time step through the RNN layer to get the hidden

State and first training (Eq. (11), (12));

Pass the hidden state through the fully connected layer to get the predicted output for this time step (Eq. (14)–(18));

Normalization data (Eq. (19)–(22));

Store the predicted output for this time step;

End while

End for

Calculate predicted QoS = fully connected layer (hidden state)

End

Algorithm 2

BWO algorithm composite the services discovered by RNN

Input: Services discovered by RNN

Output: Composite the best services

Set

//Evaluate the fitness of each spider (based on the QoS composition)

Initialize

//Select the best spiders and use their webs to attract new spiders to the colony

For ($i=1$ to n) **For** ($j=1$ to d) Initialize BWO(i,j) Eq. (23)

End for

End for

// Update the population of spiders based on the new arrivals
Fitness (BWO)

While ($t < \text{Max iteration}$) **do**

Update the number of spider

Fitness (BWO) Eq. (27)

Calculate best composition service with using Eq. (28) respect to the spider number

Update fitness function

Repeat steps until the stopping criterion are met (maximum number of iterations or satisfactory fitness level is reached)

End while

Return the best solution found

dynamic smart city scenario. The simulations were run on a Linux OS, Corie i7 Intel Processor, and 32 GB of RAM.

The service, which falls under the government and planning domain, generates a street dashboard based on inputs of street video, temperature, and water level. The service uses Dublin

TABLE II
DATA SET USED IN SMART CITIES

Parameters	Details
Location	Dublin city center
Area Covered	Approximately 2Km ²
Number of Gateways	500
Mobile Gateways	250
Static Gateways	250
Movement Patterns	Provided by cooja simulator
Number of Services	20.000, 40.000, 60.000, 80.000, 100.000
Number of Hops	1 to 5
Workload	Closed workload with citizens making a request every 6 minutes for 5 hours.
Requests	5000 requests obtained for each simulation.
Metrics	The number of rate of solved requests, Average availability, reliability cost, energy consumption and latency per request.

TABLE III
PARAMETERS USED TO TRAIN THE MODEL

Parameters	Value
Learning rate	0.01
Number of hidden Layers	3,5,7
Number of neurons per layer	50,100
Learning rate:	0.01
Batch size	32,64
Number of epochs	50,100

points of interests (POIs) and their domains as a reference. Each POI is automatically annotated with domain information based on its location and tags in open street maps. City services are defined using Dublin POIs [38] and their domains. The simulation is based on Dublin city and includes 500 gateways, of which 250 are mobile and 250 static. The experiments were conducted with different numbers of services, and various parameters were varied for each approach [38]. Table II summarizes the essential information for the setup, while Table III presents the parameters used for training.

B. Results

The study's results revealed essential insights into the performance of various algorithms used for service discovery and composition in IoT. Specifically, the developed RNN-BOW algorithm performances were compared with other popular techniques, such as GQSC, ANN-PSO, DRL, and ML techniques. In addition, the study focused on a range of performance metrics, including the rate of solved requests, average availability, reliability, cost, energy consumption, and latency. The distinct performance attributes observed in our study for the RNN-BOW, GQSC, ANN-PSO, and DRL algorithms can be traced back to their unique computational architectures and operational principles. Our proposed method, the RNN-BOW algorithm, capitalizes on the predictive prowess of RNN for forecasting the QoS and synergizes it with the BWO to enable effective service composition. This combination propels the RNN-BOW to outperform the other algorithms

on certain metrics, such as the rate of solved requests and reliability. However, our algorithm's and others' performance characteristics are influenced by the diversity and distinct QoS attributes of the IoT services in our data set. This aspect needs to be taken into account. The superior performance of the RNN-BOW on certain metrics comes with tradeoffs. While its integrated structure allows for efficient modeling and predicting of the diverse service characteristics, resulting in improved reliability and solved request rate, this comes at a cost. The increased computational demands and resources necessitated by the two-pronged RNN-BOW approach lead to a rise in energy consumption and overall costs.

Furthermore, the role of hyperparameters in the performance of the algorithms should be noted. Different settings, particularly within our proposed method's RNN and BWO components, could influence learning rates, the number of epochs, and the balance between exploration and exploitation in the BWO. These results bear practical implications for real-world IoT service discovery and composition scenarios. While the high reliability and success rate of the RNN-BOW algorithm make it a compelling choice for applications where these factors are crucial, its higher energy usage and costs must be considered. More straightforward algorithms, such as ML or DRL, could be better suited for applications where energy efficiency or cost-effectiveness are primary considerations. Therefore, understanding the inherent tradeoffs and mapping them to the specific requirements of IoT applications can guide us in making more informed algorithmic choices.

The integrated nature of the RNN-BOW increases computational demands, which results in a spike in energy consumption and overall costs. The implications of these findings extend to real-world applications of IoT service discovery and composition. RNN-BOW's high reliability and solved request rate make it a strong contender for applications where these factors are critical. But its higher energy consumption and costs may be a deterrent for scenarios where energy efficiency or cost-effectiveness are crucial, suggesting that simpler algorithms like ML or DRL could be more suitable. This underlines the importance of understanding these tradeoffs to make informed decisions when choosing an algorithm for specific IoT applications. Additionally, the effect of hyperparameters on the performance of algorithms, especially in our method's RNN and BWO components, cannot be ignored. Variations in these settings can impact learning rates, the number of epochs, and the balance between exploration and exploitation in the BWO, thus influencing the overall performance. Analyzing these metrics allowed us to identify the strengths and weaknesses of each algorithm and compare them against each other.

Fig. 3 shows the performance of different algorithms for IoT service discovery and composition, based on the percentage of successfully solved requests. The y-axis, which ranges from 0% to 100%, shows the proportion of successfully resolved requests, while the x-axis indicates the number of services. The proposed method has the best performance, with a success rate of 0%, 40%, 60%, 70%, 75%, and 80% for 20 000, 40 000, 60 000, 80 000, and 100 000 services, respectively. The "ML algorithm" has a success rate of 0%, 35%, 55%, 65%, 68%, and 72%, respectively. GQSC has lower percentage of

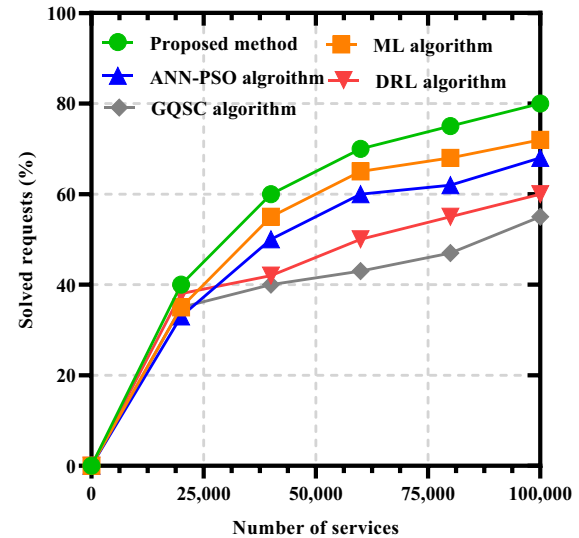


Fig. 3. Average precision in solved requests in a different number of services.

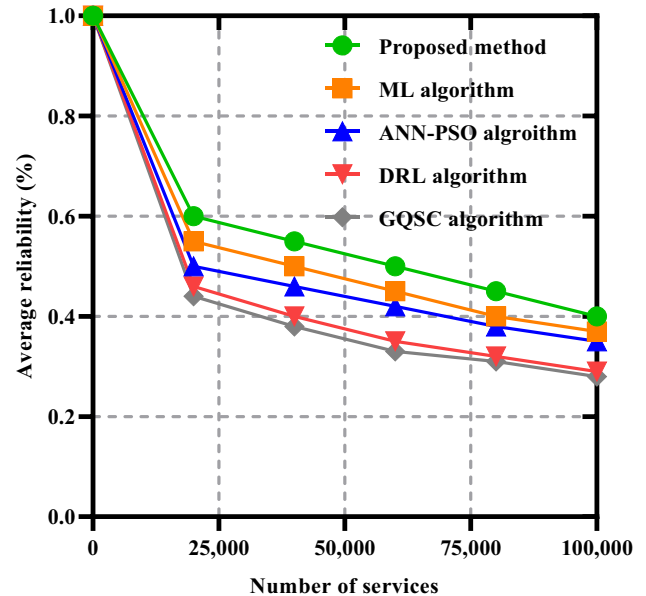


Fig. 4. Average precision in reliability in a different number of services.

solved requests in comparison with other algorithms. In this article, we used RNN algorithms for the prediction of the QoS and combine the best service with BWO algorithms in order to increase the performance of solved requests. Combining the strengths of RNN and BWO can lead to a powerful system for IoT service discovery and composition. The accurate QoS predictions from RNN can guide the BWO algorithm to composite high-quality services.

Reliability is a key factor in the successful adoption and deployment of IoT technologies, as it impacts the safety, performance, and trustworthiness of these systems. Fig. 4 depicts the reliability of the same five methods for IoT service discovery and composition as a function of the numbers of services. The x-axis shows the number of services tested, ranging from 20 000 to 100 000, in increments of 20 000. The y-axis shows the reliability score of each method, ranging from

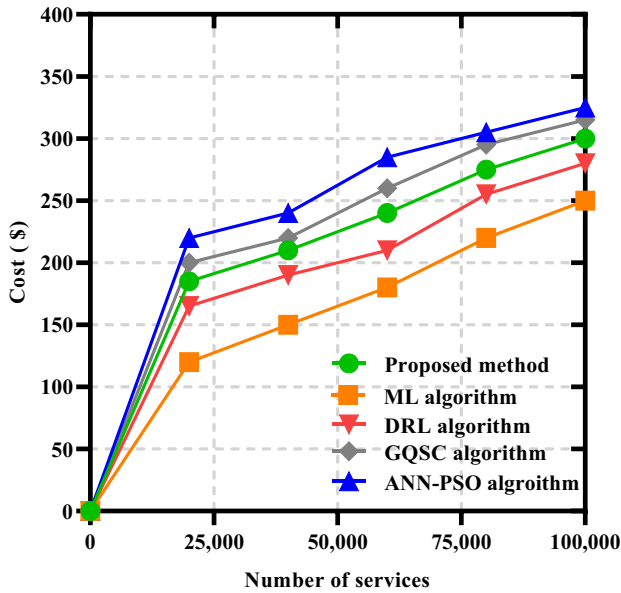


Fig. 5. Average cost for a different number of services.

1 (most reliable) to 0 (least reliable), with five additional increments of 0.2. The proposed method is the most reliable, with a score of 0.55 at 25,000 service levels. The ML algorithm, ANN-PSO algorithm, DRL algorithm, and GQSC algorithm all have decreased reliability scores as the number of services increases. The GQSC algorithm is the least reliable method at all tested service levels, with a reliability score of only 0.28 at 100,000 services.

IoT service discovery and composition costs can include both direct and indirect expenses, such as hardware and software costs, development and maintenance fees, and operational costs. Factors influencing cost include the level of interoperability and standardization among devices and services. Fig. 5 compares the five methods mentioned above with respect to the cost of service discovery and composition. The ML algorithm is the most cost-effective, with costs ranging from 0 to 250, depending on the number of services. The most expensive method is the ANN-PSO algorithm, with costs ranging from 220 to 325, depending on the number of services. Our proposal has a higher cost than the ML and DRL algorithms, because the ML algorithm and DRL only used one algorithm for discovering and composition, in our method, we combined the metaheuristic algorithm for the composition issue.

Availability is an essential factor in IoT service discovery and composition, as it determines the degree to which a service is accessible to users and devices. To ensure high availability, it is essential to design IoT service discovery and composition systems with redundancy and failover mechanisms and to monitor the system for issues that could impact availability. Fig. 6 shows the availability of IoT systems as a function of the number of services, for the same set of algorithms used for comparison purposes.

The x -axis shows the amount of services, with options ranging from 20,000 to 100,000, while the y -axis shows the availability of services, with options ranging from 0.2 to 0.5. The proposed method has the highest availability, ranging from

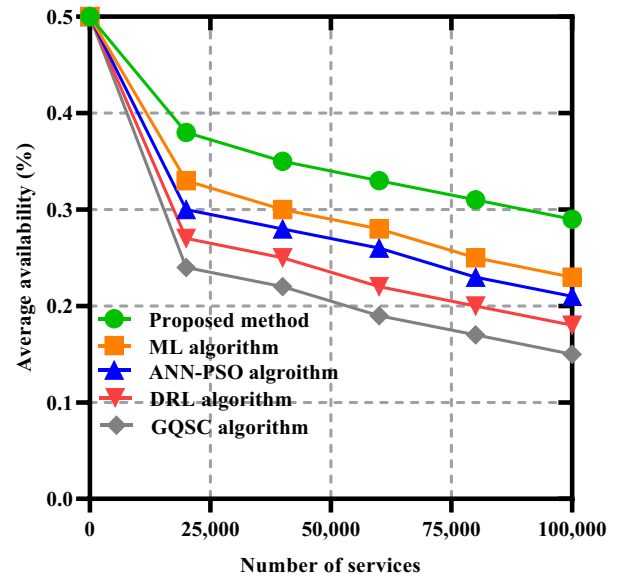


Fig. 6. Average availability in a different number of services.

0.5 to 0.29, depending on the number of services. The GQSC algorithm has the lowest availability, ranging from 0.50 to 0.15, depending on the number of services. Overall, Fig. 6 suggests that the proposed method has the highest availability and is the most reliable option for IoT service discovery and composition, particularly for larger numbers of services. However, it also highlights the importance of designing highly available systems and the need to monitor for issues that could impact availability.

Energy consumption is also essential in IoT service discovery and composition, particularly for battery-powered devices. Factors that can influence energy consumption include the complexity and frequency of discovery and composition processes, the level of optimization in the system, and the use of energy-efficient protocols and hardware. Therefore, to minimize energy consumption, it is essential to design IoT service discovery and composition systems with energy efficiency in mind and to implement strategies, such as sleep modes and energy harvesting, to conserve energy. Fig. 7 shows, comparatively, the energy consumption of IoT system versus the number of required services.

The ML algorithm has the lowest energy consumption, with energy consumption ranging from 80 to 480 kW, ranging between on the number of services. On the other hand, the proposed method has a relatively low-energy consumption, ranging from 100 to 540 kW, depending on the number of services. The ANN-PSO algorithm has the highest energy consumption, ranging from 200 to 600 kW. The Fig. 7 indicates that the ML algorithm has the lowest energy consumption and is the most energy-efficient option for IoT service discovery and composition. The energy consumption in the method proposed in this work is higher compared to ML, since the ML algorithm only use one single algorithm for the discovery and composition process, whereas our approach involved combining the metaheuristic algorithm for composition.

In IoT, service discovery refers to the process of identifying and locating services or resources available on a network.

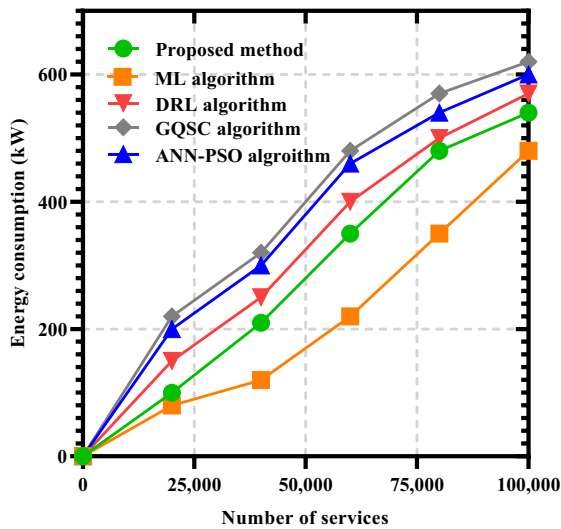


Fig. 7. Energy consumption in a different number of services.

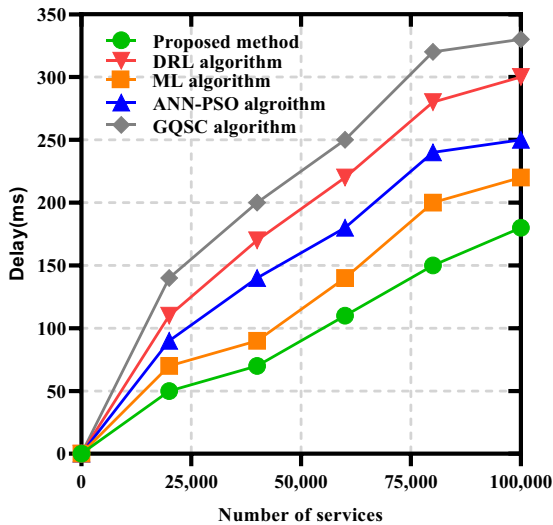


Fig. 8. Latency in a different number of services.

Latency in service discovery is the time it takes for a device or application to discover available services. Lower latency in service discovery is generally desirable, as it enables devices and applications to quickly and accurately locate available services and resources on the network. Fig. 8 shows the results of comparing different methods and algorithms for IoT service discovery and composition based on their latency performance as a function of the number of services. The method proposed in this article has the lowest latency performance across all service sizes, with latencies ranging from 50 to 170 ms. The ML algorithm also achieves similar performances, with latencies ranging from 70 to 220 ms. The ANN-PSO algorithm has slightly higher latencies, ranging from 90 to 250 ms. Finally, the DRL algorithm reaches, ranging from 110 to 300 ms.

Fig. 9 illustrates the rate at which the algorithms converge in relation to the volume of service requests. It is noticeable that the proposed approach reached convergence more swiftly than all other methods examined in our research. Furthermore,

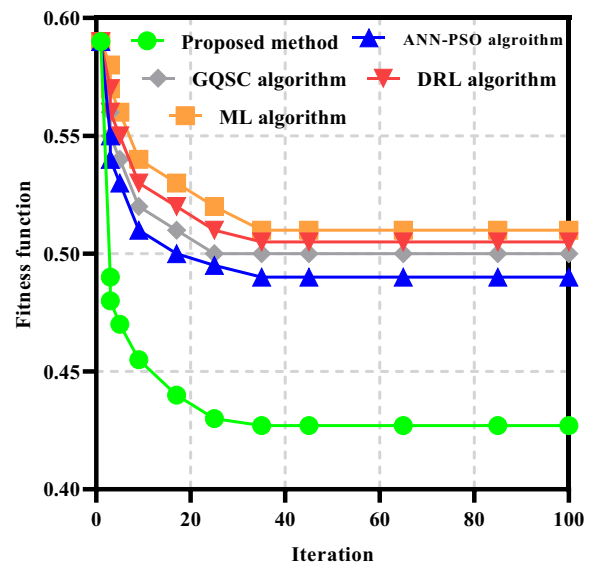


Fig. 9. Fitness function in a different number of services.

it reached its minimum value approximately after 40 iterations. Hence, the results are only shown up to the 100th iteration.

C. Discussion

The proposed methodology combines the strength of RNNs and the BWO algorithm, aiming to outperform traditional methods in satisfying user service needs within smart city networks. The RNN, using its LSTM networks, draws upon historical data to give precise QoS predictions, aiming to adhere to user preferences across a variety of network conditions. Essentially, it has the capability to manage a broad range of services, evaluate their QoS, and select the top- K services based on QoS for specific tasks. This capability is especially advantageous in resource-limited environments, aiming to reduce the overhead associated with service discovery. Following service discovery, the BWO algorithm is used with the aim to optimally composite services to meet system requirements. The BWO algorithm, modeling the behavior of a black widow spider, considers the attractive and repulsive forces between services, aiming to ensure an effective service composition strategy. The combination of RNN for service discovery and BWO for service composition allows the methodology to not only adapt to diverse network conditions but also optimally select and composite services. This aims to provide a superior, customized user experience in smart cities. However, the improved performance in these areas, which demonstrates the effectiveness of the RNN-BWO underlying mechanism, does come at a cost, both metaphorically and literally. The RNN-BWO has high-computational demands, leading to increased energy consumption and costs. The balance between the improved rate of resolved requests and reliability and the higher costs and energy consumption should be considered in real-world applications. ML algorithms, especially supervised learning algorithms, are often used in service discovery and composition. ML algorithms are often chosen for their efficiency and accuracy, but their

performance largely depends on the quality and quantity of training data available. On the other hand, DRL, a subfield of reinforcement learning that integrates deep learning, excels in situations where a clear reward signal can be maximized over time. The ANN-PSO algorithm's ability to optimize complex systems can be utilized in IoT service discovery. In contrast, the GQSC algorithm's quantum-inspired metaheuristic approach offers a unique perspective on the IoT service discovery problem. The primary objective of the study was to assess the efficiency of the RNN-BWO algorithm in IoT service discovery and composition within the context of smart cities. The results confirmed the superior performance of this model in meeting user service requirements across a wide range of network conditions. Despite its superiority, the algorithm's high-computational demands lead to increased energy consumption and costs, creating a significant tradeoff that needs careful consideration. It was also discovered that the performance of the RNN-BWO algorithm is heavily influenced by hyperparameters, indicating potential areas for future research to optimize these parameters.

V. CONCLUSION

The IoT-based smart cities can significantly enhance the quality of life for their citizens by leveraging advanced technologies, such as sensors, connectivity, and automation to optimize infrastructure, services, and sustainability. Service discovery and composition in IoT-based smart cities are crucial for meeting the changing requirements of citizens and ensuring that the services selected meet their QoS expectations. This article introduces a novel approach for solving the service composition problem in IoT using RNN and BWO algorithms. The system uses RNN-LSTM to predict the QoS of different services and, then, employs the BWO algorithm to combine them in order to satisfy the user's request. The use of RNN and BWO algorithms leads to higher accuracy in service composition, which can enhance the overall performance of IoT systems. The study evaluated a simulated smart city environment using an urban context-based approach with various parameters and performance metrics, including rate of solved requests, reliability, cost, energy consumption, and latency. The simulation results revealed that the method proposed in this work performed well for important metrics, such as latency, reliability, availability, and solved requests, indicating its effectiveness in IoT service discovery and composition. However, the method achieved higher energy consumption and cost, suggesting that more research may be needed to optimize these factors for practical implementation in real-world scenarios. In the future, it would be interesting to test the proposed approach in a real smart city environment to validate the simulation results. Additionally, deep learning techniques could be used to improve the accuracy of QoS prediction further, and metaheuristic algorithms could be explored to optimize the composition process. Finally, new metrics for evaluating energy consumption and cost could be introduced to make the proposed approach more efficient and cost-effective.

REFERENCES

- [1] A. Heidari, N. J. Navimipour, M. Unal, and G. Zhang, "Machine learning applications in Internet-of-Drones: Systematic review, recent deployments, and open issues," *ACM Comput. Surveys*, vol. 55, no. 12, p. 247, 2023, doi: [10.1145/3571728](https://doi.org/10.1145/3571728).
- [2] Z. Ali et al., "A generic Internet of Things (IoT) middleware for smart city applications," *Sustainability*, vol. 15, no. 1, p. 743, 2023.
- [3] A. Heidari, M. A. J. Jamali, N. J. Navimipour, and S. Akbarpour, "A QoS-aware technique for computation offloading in IoT-edge platforms using a convolutional neural network and Markov decision process," *IT Prof.*, vol. 25, no. 1, pp. 24–39, Jan./Feb. 2023, doi: [10.1109/MITP.2022.3217886](https://doi.org/10.1109/MITP.2022.3217886).
- [4] S. S. Sefati, A. M. Nor, S. Halunga, and O. Fartu, "A novel routing protocol based on prediction of energy consumption and link stability in mobile Internet of Thing (MIoT)," in *Proc. 25th Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, 2022, pp. 12–16.
- [5] W. Ochoa, F. Larrinaga, and A. Pérez, "Context-aware workflow management for smart manufacturing: A literature review of semantic Web-based approaches," *Future Gener. Comput. Syst.*, vol. 145, pp. 38–55, Aug. 2023.
- [6] K. Nowicka, "Smart city logistics on cloud computing model," *Procedia-Social Behav. Sci.*, vol. 151, pp. 266–281, Oct. 2014.
- [7] U. K. Lilhore et al., "Design and implementation of an ML and IoT based adaptive traffic-management system for smart cities," *Sensors*, vol. 22, no. 8, p. 2908, 2022.
- [8] A. Heidari, N. J. Navimipour, M. A. J. Jamali, and S. Akbarpour, "A green, secure, and deep intelligent method for dynamic IoT-edge-cloud offloading scenarios," *Sustain. Comput. Inform. Syst.*, vol. 38, Apr. 2023, Art. no. 100859.
- [9] R. Sharma and R. Arya, "UAV based long range environment monitoring system with industry 5.0 perspectives for smart city infrastructure," *Comput. Ind. Eng.*, vol. 168, Jun. 2022, Art. no. 108066.
- [10] S. S. Sefati and S. Halunga, "A hybrid service selection and composition for cloud computing using the adaptive penalty function in genetic and artificial bee colony algorithm," *Sensors*, vol. 22, no. 13, p. 4873, 2022.
- [11] R. Wang and J. Lu, "QoS-aware service discovery and selection management for cloud-edge computing using a hybrid meta-heuristic algorithm in IoT," *Wireless Pers. Commun.*, vol. 126, no. 3, pp. 2269–2282, 2022.
- [12] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.
- [13] V. Hayyolalam and A. A. P. Kazem, "Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103249.
- [14] A. Nauman, M. A. Jamshed, Y. Ahmad, R. Ali, Y. B. Zikria, and S. W. Kim, "An intelligent deterministic D2D communication in narrow-band Internet of Things," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, 2019, pp. 2111–2115.
- [15] V. Hayyolalam, S. Otoum, and Ö. Özkasap, "Dynamic QoS/QoE-aware reliable service composition framework for edge intelligence," *Clust. Comput.*, vol. 25, no. 3, pp. 1695–1713, 2022.
- [16] H. Deng, J. Huang, Q. Liu, C. Zhou, and J. Gao, "BGSD: A SBERT and GAT-based service discovery framework for heterogeneous distributed IoT," *Comput. Netw.*, vol. 220, Jan. 2023, Art. no. 109488.
- [17] S. D. Mohana, S. S. Prakash, and K. Krinkin, "Service oriented R-ANN knowledge model for social Internet of Things," *Big Data Cogn. Comput.*, vol. 6, no. 1, p. 32, 2022.
- [18] H. Mezni, M. Driss, W. Boulila, S. B. Atitallah, M. Sellami, and N. Alharbi, "SmartWater: A service-oriented and sensor cloud-based framework for smart monitoring of water environments," *Remote Sens.*, vol. 14, no. 4, p. 922, 2022.
- [19] A. Urbieta, A. González-Beltrán, S. B. Mokhtar, M. A. Hossain, and L. Capra, "Adaptive and context-aware service composition for IoT-based smart cities," *Future Gener. Comput. Syst.*, vol. 76, pp. 262–274, Nov. 2017.
- [20] S. A. Khan and S. Rehman, "Iterative non-deterministic algorithms in on-shore wind farm design: A brief survey," *Renew. Sustain. Energy Rev.*, vol. 19, pp. 370–384, Mar. 2013.
- [21] A. H. Bidi, Z. Movahedi, and Z. Movahedi, "A fog-based fault-tolerant and QoE-aware service composition in smart cities," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 11, 2021, Art. no. e4326.
- [22] M. Hosseinzadeh et al., "A hybrid service selection and composition model for cloud-edge computing in the Internet of Things," *IEEE Access*, vol. 8, pp. 85939–85949, 2020.

- [23] A. G. Neiat, A. Bouguettaya, and M. Bahutair, "A deep reinforcement learning approach for composing moving IoT services," *IEEE Trans. Services Comput.*, vol. 15, no. 5, pp. 2538–2550, Sep./Oct. 2021.
- [24] A. Namoun, A. A. Abi Sen, A. Tufail, A. Alshantqi, W. Nawaz, and O. BenRhouma, "A two-phase machine learning framework for context-aware service selection to empower people with disabilities," *Sensors*, vol. 22, no. 14, p. 5142, 2022.
- [25] S. Sefati and N. J. Navimipour, "A QoS-aware service composition mechanism in the Internet of Things using a hidden-Markov-model-based optimization algorithm," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15620–15627, Oct. 2021.
- [26] N. Tran, T. Nguyen, B. M. Nguyen, and G. Nguyen, "A multivariate fuzzy time series resource forecast model for clouds using LSTM and data correlation analysis," *Procedia Comput. Sci.*, vol. 126, pp. 636–645, Aug. 2018.
- [27] N. Jelodari and A. A. P. Kazem, "Black widow optimization (BWO) algorithm in cloud brokering systems for connected Internet of Things," *J. Comput. Robot.*, vol. 15, no. 1, pp. 33–45, 2022.
- [28] F. Cirillo, D. Gómez, L. Diez, I. E. Maestro, T. B. J. Gilbert, and R. Akhavan, "Smart city IoT services creation through large-scale collaboration," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5267–5275, Jun. 2020.
- [29] H. Liang, X. Wen, Y. Liu, H. Zhang, L. Zhang, and L. Wang, "Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning," *Robot. Comput.-Integr. Manuf.*, vol. 67, Feb. 2021, Art. no. 101991.
- [30] S. S. Sefati and S. Halunga, "Mobile sink assisted data gathering for URLLC in IoT using a fuzzy logic system," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, 2022, pp. 379–384.
- [31] A. Naseri and N. J. Navimipour, "A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm," *J. Ambient Intell. Humanized Comput.*, vol. 10, pp. 1851–1864, May 2019.
- [32] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4041–4049.
- [33] S. S. Sefati, M. Abdi, and A. Ghaffari, "QoS-based routing protocol and load balancing in wireless sensor networks using the Markov model and the artificial bee colony algorithm," *Peer-to-Peer Netw. Appl.*, vol. 16, no. 3, pp. 1499–1512, 2023.
- [34] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values," *Transp. Res. C, Emerg. Technol.*, vol. 118, Sep. 2020, Art. no. 102674.
- [35] R. Meyes, J. Donaue, A. Schmeing, and T. Meisen, "A recurrent neural network architecture for failure prediction in deep drawing sensory time series data," *Procedia Manuf.*, vol. 34, pp. 789–797, Jul. 2019.
- [36] X. Li, Q. Ding, and J. Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Rel. Eng. Syst. Safety*, vol. 172, pp. 1–11, Apr. 2018.
- [37] S. S. Sefati and S. Halunga, "Ultra-reliability and low-latency communications on the Internet of Things based on 5G network: Literature review, classification, and future research view," *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 6, 2023, Art. no. e4770.
- [38] C. Cabrera, G. White, A. Palade, and S. Clarke, "The right service at the right place: A service model for smart cities," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2018, pp. 1–10.



Seyed Salar Sefati (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering (software) from the Islamic Azad University of Tabriz Branch, Tabriz, Iran, in 2018 and 2021, respectively.

According to research-gate RI, he is higher than 99% of all members who first published in 2021, which is a testament to his research skills and dedication. He also published five ISI journal papers during this time, with his first paper being published in 2021. His research interests include cloud

and distributed computing, Internet of Things, software-defined networking, information systems, computational intelligence, and evolutionary computing.

Dr. Sefati was ranked among the top three students out of a total of 130 students, during his master's degree. He is currently a member of IEEE Young Professionals and develops innovative solutions for ultra-reliable low-latency communications URLLC services on the Internet of Things. He is a member of the MOTOR 5G Project, funded by the European Union's Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie Grant.



Bahman Arasteh was born in Tabriz, Iran. He received the master's degree in software engineering from the Azad University of Arak, Arak, Iran, in 2006, and the Ph.D. degree in software engineering from the Tehran Science and Research Branch, Azad University, Tehran, Iran, in 2014.

He is currently an Associate Professor with Istinye University, Istanbul, Turkey. His research interests include search-based software engineering, software testing, software fault tolerance, and optimization algorithms in computer engineering.

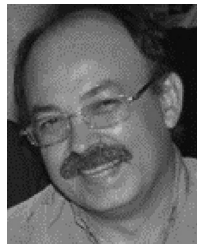


Simona Halunga (Member, IEEE) received the M.S. degree in electronics and telecommunications and the Ph.D. degree in communications from the University POLITEHNICA of Bucharest, Bucharest, Romania, in 1988 and 1996, respectively.

From 1996 to 1997, she followed post-graduate courses in management and marketing organized by the Romanian Trade and Industry Chamber and the University POLITEHNICA of Bucharest in collaboration with Technical Hochschule Darmstadt, Darmstadt, Germany. From 1997 to 1999, she was

a Visiting Assistant Professor with the Electrical and Computer Engineering Department, University of Colorado at Colorado Springs, Colorado Springs, CO, USA. Since 2006, she was a Full Professor with Telecommunications Department, Faculty of Electronics, Telecommunications and Information Technology, University POLITEHNICA of Bucharest. She has published over 180 papers in different scientific journals or in the proceedings of different scientific conferences and participated in several nationally funded, NATO Science for Peace, and European research projects. Her research interests include multiple access systems and techniques, digital communications, communications systems, and digital signal processing for telecommunications.

Prof. Halunga was the Director of the Scalable Radio Transceiver for Instrumental Wireless Sensor Networks-Sarat Nationally Funded Research Project from 2012 to 2016.



Octavian Fratu received the Ph.D. degree in electronics and telecommunications from the University POLITEHNICA of Bucharest, Bucharest, Romania, in 1997.

He achieved postdoctoral stage as a Senior Researcher in 3rd Generation Mobile Communication Systems, based on a research contract between CNET-France, ENS de Cachan, Cachan, France, and Universite Marne la Vallée, Marne-la-Vallée, France. He is currently a Professor of Electronics, Telecommunications and Information

Theory Faculty. His publications include more than 200 papers published in national or international scientific journals or presented at international conferences. His research interests include digital mobile communications, radio data transmissions, mobile communications, and wireless sensor networks.

Prof. Fratu participated as the Director or Collaborator in many international research projects, such as "Reconfigurable Interoperability of Wireless Communications Systems (RIWCoS)" (NATO Science for Peace Research Project, 2007–2010), "REDICT—Regional Economic Development by ICT/New media clusters" (FP7 CSA Project, 2008–2009), "eWALL for Active Long Living-eWALL" (FP7 IP Project, 2013–2016), and others. He is currently leading the local team involved in RECOMBINE MSCA-H2020-RISE Project and he is also involved in MOTOR-5G MSCA-H2020-ITN Project.

Asgarali Bouyer received the Ph.D. degree in computer science from Universiti Teknologi Malaysia, Johor Bahru, Malaysia, in 2011.

He is currently an Associate Professor with the Faculty of Computer Engineering and Information Technology, Azarbaijan Shahid Madani University, Tabriz, Iran, and the Department of Software Engineering, Faculty of Engineering and Natural Science, Istinye University, Istanbul, Turkey. His research interests include complex networks, social media mining, data clustering, and resource manage-



ment systems in cloud computing.