IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 2, April 2026, pp. 1097~1108
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i2.pp1097-1108

Intelligent self-organizing microservice composition using hybrid learning for neonatal ward

Sharon Poornima, Ashok Immanuel V
Department of Computer Science, CHRIST (Deemed to be University), Bengaluru, India

Article Info

Article history:

Received Aug 5, 2025
Revised Jan 5, 2026
Accepted Jan 22, 2026

Keywords:

Critical care
Health care
Hybrid learning
Microservice composition
Self-organization

ABSTRACT

This research presents an innovative self-organizing microservice composition model specifically tailored for dynamic and time-sensitive healthcare environments such as neonatal intensive care units (NICU). A hybrid machine learning classifier detects neonatal conditions and assigns treatment plans based on real-time vitals. The composition process is guided by a deep learning agent that combines unsupervised and reinforcement learning to develop intelligent bonding strategies. Microservices act as autonomous agents, supporting decentralized service choreography within the self-organizing framework. The bonding strategies of direct bonding and shared bonding are implemented for single conditions and coexisting conditions, respectively. The simulation results are based on actual NICU data, demonstrating the ability of the model to dynamically compose services while ensuring optimal resource utilization. The model demonstrates an adaptive and dynamic composition through emergence and continuous learning for changing clinical conditions, and demonstrates emergent behavior through reinforcement learning. The model's predictive capabilities enable anticipatory service loading, providing context-aware treatment in critical healthcare scenarios. This self-organizing architecture model offers a scalable and robust solution for autonomous, decentralized service choreography in critical healthcare environments.

Corresponding Author:

Sharon Poornima
Department of Computer Science, CHRIST (Deemed to be University)
Bengaluru, India
Email: sharon.poornima@res.christuniversity.in

1. INTRODUCTION

The internet of things (IoT) has gained significant traction in the past decades owing to its ubiquitous nature and ability to connect multiple devices for seamless proliferation of information through networks, making real-time and remote applications more accessible. The usage of IoT devices to monitor and analyze data in healthcare has been a boon to healthcare professionals as it solves two important problems in the current healthcare system, improving accessibility to healthcare services and reducing costs [1]. Service oriented computing has enabled pervasive devices such as sensors, actuators and other peripherals to virtualize the data and functionality provided by them to be accessed as services via cloud [2]. Microservices Architecture has evolved into a technology through which services are provided by devices commonly termed as IoT. Microservices architecture is preferred to service oriented architecture as it provides more scalability, versatility and performance, especially in healthcare applications [3]. Microservices are loosely coupled and are the best fit to represent resource-constrained devices, which are typically used in IoT. Each sensor or IoT device can provide atomic or composite services based on its functionality, and the failure of a single microservice does not cause the failure of the entire system, as in the case of monolithic architecture [4]. The atomic services cannot always meet the requirements of the user, and multiple services are combined to provide the required functionality to the user. Combining the atomic services to form a composite service is termed as service composition [2], [5]. The IoT environment is highly dynamic and open, and poses three major challenges, firstly there is network instability and dynamism due to the mobile nature of the devices that may leave and reconnect to the network due to poor network connections and other instability issues [6]. A second challenge is the increase in complexity of the service composition process, proportional to the increase in the number of devices and services provided by these devices [7]. A third challenge is functional scalability; each device can provide multiple services, and there could be multiple devices providing services with similar functionality, causing a combinatorial explosion [2].

Traditional centralized service composition techniques are insufficient for applications in highly dynamic and open environments such as the IoT. There is a need for decentralized models that can effectively adapt to these highly dynamic and open environments. Self-organization is a well-known bottom-up approach that supports the highly dynamic and open nature of the IoT environment [8]. This research proposes a self-organizing microservice composition model for the IoT ecosystem to facilitate service composition in these complex environments. Some of the features of self-organizing service composition are that the interactions among the services are decentralized and local, enabled by bio-inspired mechanisms, without the need for a centralized controller, and the service composition is dynamic (at runtime), and adapts to service failures. There is autonomous and spontaneous interaction among the services, enabling improved response and scalability, and support for emergent behavior, i.e., a composite service is formed through the bio-inspired mechanisms and continuous learning. The services also autonomously reorganize in response to changing environmental conditions and workload variations [9]. The research presents a decentralized and autonomous microservice composition model self-organizing microservices using intelligent learning (SMIL) model that enables decentralized and autonomous microservice interactions through bio-inspired mechanisms for adaptive and self-organizing service composition. The central component of this model is the self-organizing BIOCORE, which is developed and implemented through a self-organizing map (SOM) to emulate the bio-inspired mechanisms of spreading, evaporation, aggregation, and bonding to enable emergent self-organization. This unsupervised learning approach is combined with a deep learning agent that enables the model to compose valid service sequences that adapt dynamically at runtime. By focusing on these dimensions, this research aims to enhance response time, minimize service failure, and enhance neonatal healthcare outcomes. The applicability of this work is highlighted by a recent UN report [10], which positions India among the top five nations with the highest number of preterm births, reflecting the compelling need for efficient and innovative neonatal care solutions.

The rest of this paper is divided as follows: section 2 delves into the literature overview on self-organization and recent service composition models and frameworks. Section 3 describes the self-organizing microservice composition model, its components, and the service composition process. Section 4 explains the experimental setup, simulation outcomes, and performance evaluation. Section 5 concludes the paper and suggests possible future enhancements. This research makes an IoT-driven healthcare contribution by introducing a scalable, adaptive, and intelligent service composition model, ultimately improving neonatal healthcare outcomes and IoT-based healthcare services.

2. RELATED WORK

The nature of an IoT environment is inherently open and dynamic, making it vulnerable to network instability and potential service failure. Additionally, when the services are composed (added into the workflow) during design time, some of the selected services may not be available during runtime, leading to the complete failure of the composition. In contrast, self-organizing service composition dynamically composes services that are provided by IoT devices, suiting the open nature of the IoT system. The advantages of self-organizing service composition are that it enables adaptive service composition, enhancing scalability. It improves the service execution, especially in dynamic environments, ensuring efficient and reliable performance. Self-organizing mechanisms mimic biological and physical processes that are observable in nature. Self-organization has been applied to a multitude of technological applications. With the rise of cyber-physical and automated systems, research in self-organization has gained much interest. However, the implementation of self-organization in service composition has not yet reached its full potential. An initial survey of self-organization and service composition has given insights that methods for self-organization and its patterns have existed in technological applications for a few decades.

A literature survey was conducted using the keywords “self-organization,” “self-organizing,” “self-evolving,” “self-adaptive,” “service composition,” and “self-composition.” Based on this search, 50 relevant papers published between 2014 and 2024 were identified and analyzed. The methodologies employed for developing self-organizing systems are illustrated in Figure 1.

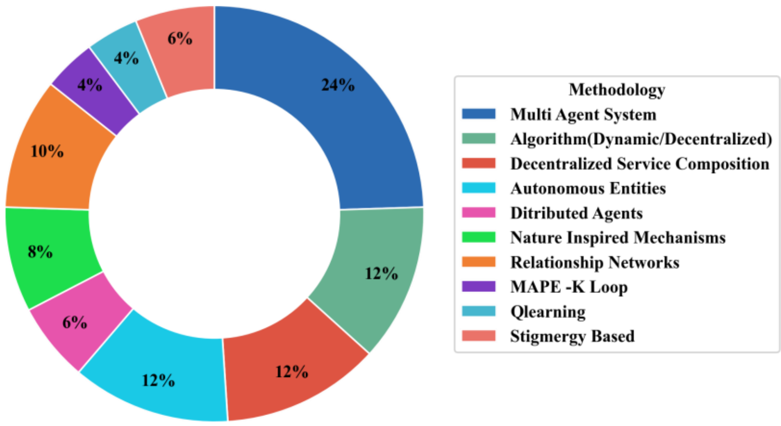

Figure 1. Methodologies used in self-organization

In recent years, the research has pointed towards the increase in pervasive applications, particularly using the IoT, where self-organization has emerged as the apt methodology to develop these applications. Different frameworks, models, and algorithms have been studied and reviewed to deepen the understanding and knowledge of self-organization in system development. Some of the research that is most cognizant and recent with self-organization and service composition has been tabulated in Table 1.

Table 1. List of self-organizing systems

No.	Research focus	Key contributions	Limitations
1.	Goal-oriented approach for service composition in mobile computing environments [6]	Introduced GoCoMo, a goal-driven service composition model for dynamic IoT environments	Limited scalability
2.	Integrated agent-based model for adaptive service composition in networked systems [10]	Developed an agent-based approach for self-evolving service composition	Suffers from cold-start issues
3.	Coordination model using learning techniques for on-the-fly composition of robust services [11]	Applied Q-learning for adaptive service composition in the SAPERE framework	High computational cost
4.	Semantic matching-enabled learning-based model for dynamic service composition [12]	Integrated semantic matching into a learning-based coordination model for IoT services	Risk of network flooding due to excessive information dissemination
5.	Bio-inspired emergent strategy for self-organization in autonomous IoT environments [13]	Used neurotransmitter-based communication for cooperative IoT node interactions	Lacks robustness in dynamic environments
6.	Self-advisory mechanisms in self-organizing systems based on emergent behavior [14]	Introduced self-advising mechanisms to enhance context-awareness, validated through experiments in the NASA-ANTS mission	Scalability challenges in large systems
7.	Multi-agent system for coordinated self-organization of autonomous shuttle fleets [15]	Developed an agent-based framework to optimize autonomous shuttle fleet operations	Lacks real-time dynamic operations for ride-sharing
8.	Resilient mobile service composition via cooperative agent-based communities [16]	Introduced SBOTI, a stigmergic-based optimization model for decentralized service composition	Limited adaptability to changing environments
9.	Cross-layer design for self-organizing and adaptive communication in IoT networks [17]	Developed 2SAEC-IoT for efficient self-organizing and self-configuring IoT communications	Increased complexity for resource-constrained devices
10.	IoT recommendation system based on a self-organizing multi-agent framework [18]	Proposed NARIoT, a self-organizing multi-agent recommendation platform for IoT environments	Lacks advanced intelligence and networking capabilities

The literature review emphasizes the increasing significance of self-organizing service composition in dynamic environments. Several approaches, ranging from multi-agent systems, decentralized algorithms, to nature-inspired methods, have been explored. Self-organizing service composition is highly suitable for IoT-based healthcare environments, especially in critical settings like the neonatal intensive care units (NICU). It supports adaptive, decentralized coordination of services, increasing real-time monitoring, resource use, and fault tolerance. However, despite the potential of self-organization to provide efficient and reliable composition of services, its full application in healthcare remains under-explored, necessitating the need for further research in this domain. The contributions of the proposed model in comparison with the existing approaches have been presented in Table 2.

Table 2. Contribution of the proposed model vs. existing models

Feature	Existing models	Proposed model
Learning approach	Pure reinforcement learning is used in [10], [11]. Rule-based systems such as [6] face cold-start and slow convergence.	Hybrid learning: combines unsupervised learning and reinforcement learning enabling faster convergence and adaptive bonding.
Emergence and self-organization	Emergent behavior is discussed conceptually in [13], [14]. There is no real-time learning or runtime evaluation.	Formation of effective emergent patterns aided by SOM and reinforcement learning at runtime.
Communication and nature-inspired optimization	Spreading causes the risk of network flooding (observed in [11], [12]). Stigmergic optimization is used in [16]. These strategies are not applied to healthcare.	The ant agent uses pheromone strategies to optimize bio-inspired mechanisms, reducing network and computational overload.
Agent-based architecture	Multi-agent systems are used for distributed control [10], [15], [16], [18].	Microservices collaborate for learning-based, autonomous, self-organizing service composition.
Fault tolerance and re-bonding	If the service compositions are fragile, failures can cause a complete breakdown [6], [10].	The fallback mechanism is implemented during bonding. Re-bonding of failed services is enabled using a memory buffer to ensure graceful recovery.
Service aggregation and composition	Services are often invoked in isolation, leading to multiple instances of the same service [11], [12].	Aggregates overlapping services across treatment plans, reducing redundancy and enabling decentralized composition.
Monitoring and adaptation loop	Few systems like [17] explicitly model the monitor–analyze–plan–execute–knowledge (MAPE-K) loop.	Agents implicitly perform the processes involved in a MAPE-K cycle.

3. SELF-ORGANIZING MICROSERVICE COMPOSITION MODEL

Self-organization is a well-known bottom-up approach that can be used to solve the inherent problems in an IoT system [8]. This research proposes modelling using agent-based systems. Multi-agent systems are a precise match to represent the self-organizing microservice composition model. Several properties of multi-agents make this method most suitable to be utilized in the proposed model. Multi-agents emulate intelligence as they learn continuously from the environment and are well-suited for automated applications. They are autonomous and capable of making reactive or proactive decisions based on the environment. Multi-agents also make collective collaborations by communicating with other agents in the environment and provide services to the end user [19]. In an IoT environment, microservices are touted as the appropriate architecture to represent the services and data provided by the IoT devices. Microservices can function well as agents in IoT systems as they have many similarities to agents; they are distributed, modular, autonomous, and can collaborate with other microservices and are capable of learning and adaptation, also due to their proactive or reactive properties [20]. In this model, microservices are deployed as agents to represent several functionalities of the model. The motivation for using microservices as loosely coupled agents in this model is their ability to function in a heterogeneous environment such as an IoT ecosystem, enhancing the scalability and adaptability of service composition methods.

3.1. Microservice repository in the self-organizing model

The microservice repository registers and manages available services. It ensures that services are discoverable, composable, and fail-safe. New services can be added dynamically to the repository, and the active services can bond based on the condition and shared vitals. If an active service fails, it is replaced with a similar one. An example representation of a microservice in the microservice repository is given as follows.

```
{
  "Name":  "Oxygen Delivery",
  "Type":  "Respiratory",
  "Required vitals":  ["FiO2", "SaO2"],
  "Execution time": (0.6,0.9), # in milliseconds
  "Failure probability": 0.05 , #5% failure rate
  "Alternatives":  ["Nasal Prong Oxygen", "Ventilation Setup"]
}
```

The microservice registry registers all the available services and coordinates them. During the service composition process, the deep learning agent queries the microservice registry and selects individual Microservices or composite service patterns based on the detected neonatal conditions. These services are composed using adaptive learning of emergent patterns, powered by the hybrid agent (deep learning and

SOM) integrated in the model. The composition of the services is further facilitated by service choreography, which aids self-organization when compared to service orchestration [21].

3.2. Components of the microservice composition model

The self-organizing microservice composition model has been explained in detail in a previous research article [20], and a brief description of the model has been provided below. The model's components are:

i) A machine learning-based classifier, integrating random forest (RF) and XGBoost, has been employed to accurately classify various neonatal conditions and recommend corresponding treatment plans. The model is evaluated using a real-world neonatal physiological dataset collected retrospectively under ethical clearance from the NICU of SKS Hospital and Postgraduate Medical Institute, Salem, India. The dataset contains continuous second-wise data of key physiological parameters such as heart rate (HR), blood pressure (BP), respiratory rate (RR), oxygen saturation (SaO_2), oxygen concentration (FiO_2), and temperature (Temp) recorded during the first 8 hours of life from ten mechanically ventilated neonates, comprising over 250,000 timestamped records. The complete description of the hybrid model (RF+XGBoost) is discussed extensively in a separate paper. While that article focuses on the working of the machine learning model to classify conditions and recommend treatments, the present work builds upon it by detailing the self-organizing service composition process that is triggered once a neonatal condition is detected.

ii) The BIOCORE agent comprises the microservices of spreading, aggregation, evaporation, and bonding, aiding self-organization. These microservices are the basis of the self–organizing rules that govern the system. These microservices react based on the data (neonatal vitals) that have been classified as conditions.

iii) A deep learning agent that enables bonding of relevant services, while learning from correct and incorrect bonding and adapting dynamically to service failures. The deep learning agent is supported by unsupervised learning through the SOM. SOM identifies patterns (clusters) that can be reused, providing emergence and adaptability.

iv) An ant agent will be integrated into the model to optimize the bio-inspired mechanisms using pheromone-based strategies. It will efficiently propagate information to relevant services, aggregate functionally similar microservices, and eliminate outdated service paths. This minimizes network bottlenecks, reduces search time, enhances scalability, and ensures continuous adaptation to dynamic IoT environments, collectively improving the efficiency of service composition.

v) An optimization agent that selects the optimal microservice compositions by minimizing execution time, reducing cost, and maximizing reliability. It uses a multi-objective approach (NSGA-II+HC) to ensure efficient, scalable, and quality of service (QoS)-aware service composition in dynamic IoT healthcare environments.

The system architecture of the self-organizing microservice composition model is depicted in Figure 2.

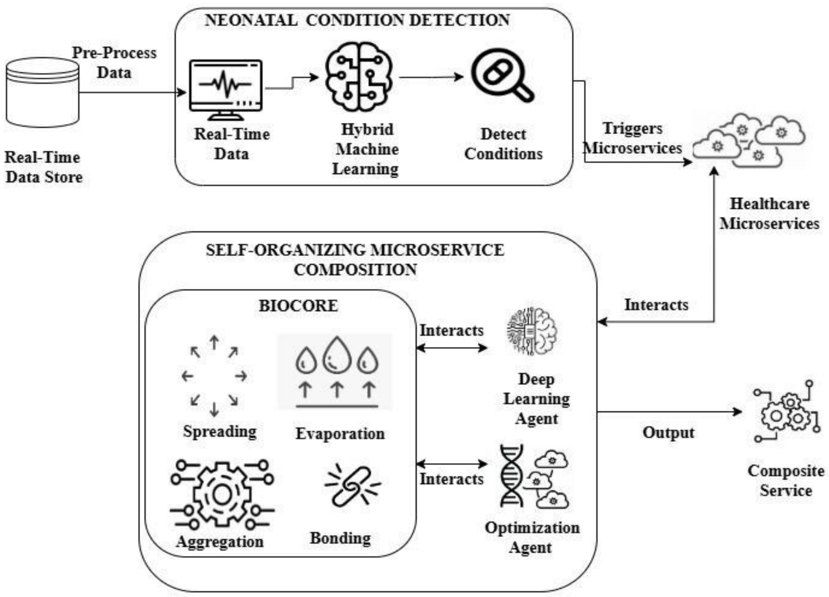

Figure 2. System architecture (self-organizing microservice composition model)

3.3. Self-organizing BIOCORE

The proposed self-organizing microservice model integrates the BIOCORE with services such as spreading, evaporation, bonding, and aggregation, and an ant agent to optimize these services, while this approach supports decentralized interactions, emergence, and self-organization. SOM provides support to the BIOCORE as a robust backbone, enabling the discovery of emergent service groupings [22]. SOM is used to cluster the services that have common vitals. This cluster of services corresponds to a particular treatment plan and is trained to automatically create bonds among the services (in the cluster). SOM is also utilized to cluster the services for multiple treatment plans to enable context-aware service sharing. Along with SOM, a graph-based rule engine is used to generate valid service sequences for the treatment plan, enabling the emergence and validity of the composed service. The SOM is trained on binary-encoded vectors of service dependencies, based on shared vitals by the services.

This unsupervised learning process enables the emergence of initial service groupings, which serve as candidate building blocks for more complex compositions. Service sequences are generated using initial service grouping primarily to support self-organizing microservice composition. The generated sequences consist of atomic services, where each sequence has one service, and composite services, where the service sequence is formed by a set of contextually related services. A directed service graph is used to expand the paths based on the treatment plans, ensuring medical relevance. Breadth-first search is applied to the graph to generate valid treatment sequences. A list of known invalid or risky sequences is maintained to remove clinically unsafe or logically inconsistent paths. The service sequences are further classified to provide treatments according to the severity of the condition.

3.4. Operational overview of the deep learning agent

Deep learning has been widely used for dynamic and adaptive service composition [23]. The agent integrates unsupervised learning and reinforcement learning to create an adaptive composition model suitable for the dynamic and uncertain environment of neonatal intensive care. Initial learning by the deep learning agent is enabled by unsupervised learning, with the help of the SOM clustering the services based on the common vitals for a condition. The graph-based rule engine generates valid sequences for a particular treatment plan. These valid sequences are used by the deep learning agent to dynamically compose services by predicting the most appropriate next microservice at each step. As successful compositions are executed, reinforcement learning takes over by rewarding the correct composition sequences (valid sequences) for the treatment plan and penalizing the invalid ones. Emergent behavior is a natural outcome of self-organization. The deep learning agent uses reinforcement learning to evaluate the effectiveness of newly discovered emergent and valid service sequences that are not explicitly defined. If an incorrect sequence is detected, the service combination is flagged as incorrect and is replaced with a valid emergent or rule-based sequence retrieved from the memory buffer. This method ensures continuity in the service compositions. Positive rewards are assigned for better outcomes driven by emergent behavior, and the internal policy of the system is updated to evolve dynamic composition strategies at runtime. Unsupervised learning is employed to discover emergent service sequences for treatment plans, while reinforcement learning is used to refine the selection and bonding of valid and optimal service sequences. The system progressively learns effective service composition patterns through rewards and penalties. The parameters used by the deep learning agent are tabulated in Table 3. The learning parameters are of the deep learning agent are chosen based on previous research in [11].

Table 3. Learning parameters of the deep learning agent

Parameter	Value	Description
Alpha	0.3	Learning rate
Gamma	0.9	Discount factor for future rewards
Epsilon	0.2	Initial exploration rate (ϵ-greedy policy)
Epsilon decay	0.995	Gradual shift from exploration to exploitation
Replay Memory	Deque (maxlen =1,000)	Stores past experiences for stable training
Optimizer	Adam	Adaptive optimizer for backpropagation
Loss function	Mean squared error (MSE)	Measures error between predicted and target Q-values

3.5. The role of the ant agent

The ant agent optimizes the services provided by the BIOCORE, namely spreading, aggregation, evaporation, and bonding. Adaptive tuning of the parameters of the ant agent [24] aids the optimization of these patterns. The proposed model incorporates the BIOCORE and ant agent, which were intended to facilitate bioinspired mechanisms such as spreading, evaporation, aggregation, and bonding. The complete model integrating the BIOCORE and ant agent optimization is under development. The current model implementation deploys a SOM to fulfil several of the intended functionalities of the BIOCORE and ant agent. The SOM now serves as a foundational building block, informing the structure and behavior of emergent service compositions. This intermediate implementation provides a critical basis for the ongoing integration of the BIOCORE and ant Agent into the complete proposed model.

4. EXPERIMENTAL SETUP AND OUTCOMES

This model is especially developed for suggesting treatment plans based on the conditions detected in the neonates. The model can detect multiple co-occurring conditions, aiding physicians in deciding critical care treatment plans based on changing clinical conditions. The model is trained and tested on several treatment plans. The sample size was selected to ensure sufficient representation of both individual and coexisting neonatal conditions. Approximately 2,000 samples were used for training individual conditions, and 3,500 samples were used for training each multiple condition, thereby enabling improved learning of complex service bonding patterns. Two commonly occurring conditions are described below, along with the process of self-organizing service composition.

4.1. Illustration of direct bonding in action

When a single condition occurs, direct bonding takes place. The continuously monitored vitals trigger an alert when the threshold values are breached. As an example, consider the following scenario. The Vital sign reading of FiO_2 is 55%, causing the condition to be detected as: C1: mild respiratory distress.

All other vital sign readings are in the normal range. In condition C1, the dominant vital is FiO_2, making it the central reference point for service bonding. The treatment plans for condition C1 involve the composition of several microservices that are already registered in the microservice repository. An instance of these services is used for the service composition. The services required for the treatment plan (T1) are:

i) Monitoring–continuously observes FiO_2 levels and other vital parameters.
ii) Oxygen delivery–increases the FiO_2 levels to enhance the oxygen content in inhaled air.
iii) Sedation–minimizes discomfort and prevents distress during the administration of respiratory interventions.
iv) Intubation-establishes a secure airway when non-invasive oxygen delivery methods are insufficient
v) Ventilation setup–configures ventilator parameters to deliver consistent and controlled respiratory support.

These services form the bonds enabled by initial unsupervised learning and deep learning to refine the bonding. Unsupervised learning identifies the initial service set (e.g., monitoring and oxygen delivery) based on clusters formed by the SOM, and deep learning identifies the optimal service sequences for a treatment plan. The service composition graph as shown in Figure 3 illustrates the order in which the services are composed for the treatment (T1). The monitoring service is always active, triggering the other services as and when required. The dashed line between (oxygen delivery and sedation) indicates that the sedation service, followed by intubation and ventilation setup, will be invoked only if the condition escalates. The patient receives the treatment plan (T1), which is classified as mild and severe based on the FiO_2 levels. The service sequence for the condition C1 and treatment plan (T1) is as:
Mild: monitoring→oxygen delivery.
Severe: monitoring→oxygen delivery→sedation→intubation→ventilation setup.

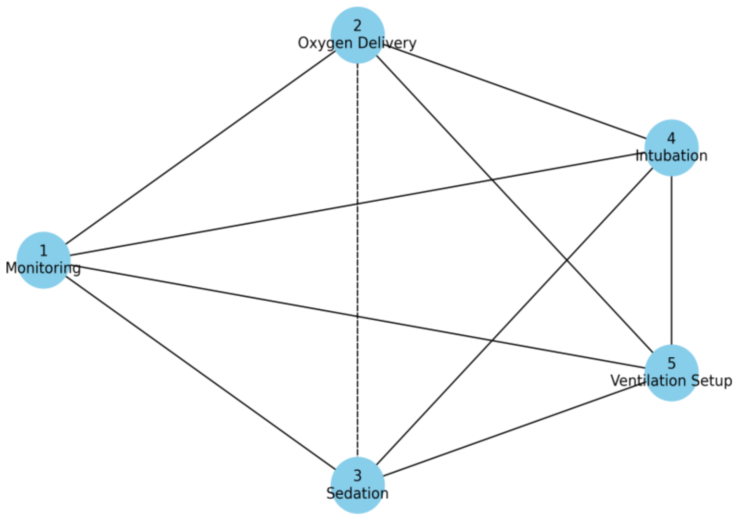

Figure 3. Service composition graph (direct bonding)

4.2. Shared bonding and aggregated service composition

In neonatal critical care, a single patient may exhibit multiple coexisting conditions that share vital indicators. One frequently occurring coexisting condition includes:

C1: mild respiratory distress (triggered when $FiO_2 \geq 30\%–60\%$).

C4: mild hypoxemia (triggered when $SaO_2 \leq 90\%$).

Figure 4 illustrates the service composition graph of treatment plans (T1+T4) for the coexisting conditions of (C1 and C4). The blue nodes indicate the services that are composed as part of the primary intervention treatment plan, and the red nodes indicate that more services have been added to the composition when the condition escalates. The dashed lines indicate from (oxygen delivery to sedation) indicates that the condition of the infant has rapidly deteriorated, requiring immediate critical care. If the condition is stable, then the respiratory support services are composed. If the condition worsens despite respiratory support, the services (intubation, ventilation support, ventilation control, and critical care) are dynamically added to the service composition. This adaptive behavior is enabled by mechanisms of emergence and deep learning and ensures timely and appropriate critical care for the infant. The treatment plan (T1+T4) is composed based on the condition of the patient. The service sequences for mild and severe treatment plans are as follows:

Mild: monitoring→alert notification→oxygen delivery→respiratory support.

Severe: monitoring→alert notification→oxygen delivery→respiratory support→sedation→intubation→ ventilation setup→ventilation control→critical intervention.

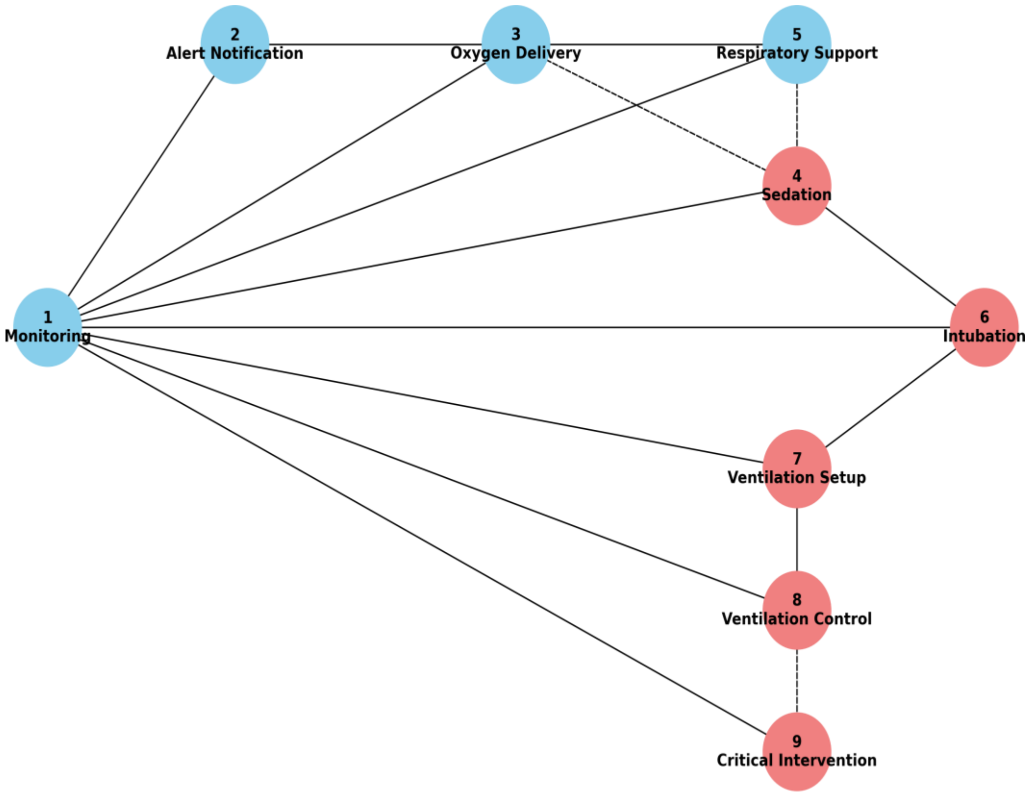

Figure 4. Service composition graph (shared bonding)

Both conditions are respiratory problems and require immediate but non-invasive intervention. The treatment plan combines the treatment labels T1 and T4. The shared and unique services for the combined treatment plan (T1+T4) are summarized in Table 4. Shared bonding is implemented, and common microservices specified as shared as shown in Table 4 are instantiated only once and used for both conditions. Unsupervised learning detects these overlaps by recognizing common services across clusters formed by the SOM. Some services are required only for condition C1, and these services are instantiated and composed based on the severity of the condition (C1). The deep learning agent is trained to compose treatment plans (T1+T4) for both mild and severe conditions of the vitals.

Table 4. Shared and unique services of shared bonding

Microservice	Shared/unique	Purpose
Vital monitoring	Shared	Adjusts dynamically to monitor relevant vitals based on condition
Alert notification	Shared	Notifies caregivers of abnormal SaO_2 or FiO_2 thresholds
Oxygen delivery	Shared	Elevates FiO_2 to improve SaO2
Respiratory support	Shared (context-aware)	Provides non-invasive breathing assistance (used differently based on condition)
Ventilation setup	Shared	Configures mechanical ventilation
Ventilation control	Shared	Adjusts breathing parameters
Critical intervention	Shared (context-aware)	Provides emergency stabilization during severe respiratory failure or rapid deterioration
Sedation	Condition-specific (C1)	Keeps the infant calm during respiratory care
Intubation	Condition-specific (C1)	Ensures a secure airway when non-invasive ventilation fails

4.3. Simulation of microservice compositions

This section presents the simulation outcomes of the proposed self-organizing microservice composition model. The model is implemented using an agent-based logic embedded with dynamic bonding mechanisms. The model simulates both direct bonding and shared bonding behavior through emergence and learning.

4.3.1. Microservice composition and execution time

A sample size of approximately 2,000 neonatal cases having the condition C1 alone was used to train the model, ensuring sufficient learning of the service composition patterns for that condition. Subsequently, the model was trained to compose services for the co-existing conditions of C1 and C4 with an expanded sample size of 3,500. The samples were extracted from the real-time dataset (obtained from the hospital) to train the model, and the service composition patterns were observed. The execution time for each service is randomly assigned, ranging from 0.1 to 3 ms, based on the expected complexity of the service and previous research related to minimizing execution time [25].

For instance, alert services were assigned shorter execution times, while critical procedures such as intubation were assigned longer execution times, reflecting the procedural demands in the practical healthcare environment. It is important to distinguish between service execution time and service composition time. Service execution time refers to the duration taken by a specific service to complete its operation, whereas service composition time refers to the time required to discover, select, and compose a set of services into a composite service.

4.3.2. Service composition for direct bonding

The deep learning agent was trained to compose service sequences to provide the treatment plan (T1) for both mild and severe cases. The service composition time and the average execution time for each microservice were recorded across 2,000 treatment plans for the patients. The average execution times were separately tracked for the mild treatment plan (T1-mild) and the severe treatment plan (T1-severe). The Services, such as sedation, intubation, and ventilation setup, are not instantiated for the mild treatment plan. Table 5 summarizes the details of the composed treatment plans.

Table 5. Average composition time (ms):-direct bonding

Microservice	Execution time (ms)	T1-mild (ms)	T1-severe (ms)
Monitoring	0.346	0.337	0.357
Oxygen delivery	0.751	0.759	0.742
Sedation	1.235	–	1.235
Intubation	2.338	–	2.338
Ventilation setup	3.000	–	3.000
Average composition time (ms)	5.173	3.000	10.500

4.3.3. Service composition for shared bonding

The deep learning agent was trained on the combined treatment plan (T1+T4), learning optimal service sequences. Shared services are instantiated once, and the execution time of these services is computed using aggregation rules based on the usage of the service as a shared or condition-specific service. The results of the composed treatment plans for the combined condition (C1+C4) are detailed in Table 6. The deep learning agent was trained on mild and severe cases separately to enhance the training speed, especially in cases of shared bonding, as the treatment plan for severe cases involves a longer composite service.

Table 6. Average composition time (ms): -shared bonding

Microservice	Execution time (ms)	(T1+T4)-mild (ms)	(T1+T4)-severe (ms)
Oxygen delivery	0.747	0.712	0.778
Monitoring	0.358	0.375	0.345
Alert notification	0.195	0.233	0.164
Respiratory support	1.527	0.914	0.911
Critical intervention	1.336	–	1.527
Sedation	2.900	–	1.336
Ventilation setup	2.900	–	2.900
Intubation	2.245	–	2.245
Ventilation control	1.160	–	1.160
Average composition time (ms)	8.635	3.578	12.773

4.3.4. Microservice utilization and bonding mechanism

The frequency of microservice usage for the two microservice bonding strategies illustrated in Figure 5 highlights the prominent role of foundational services and the benefits of context-aware service inclusion. Services such as monitoring and oxygen delivery are used more often due to their critical relevance in both direct and shared bonding. In shared bonding for conditions (C1 and C4), services like sedation and intubation were selectively bonded in severe cases of condition (C1). Additionally, several other services were aggregated and composed, reflecting runtime service dynamicity and adaptation. The simulation results demonstrate the effectiveness of the proposed self-organizing microservice composition model to dynamically compose services and adapt to varying clinical conditions in a dynamic IoT ecosystem through self-organization and pattern learning using deep learning. These promising outcomes highlight the potential of the model for deployment in remote monitoring and real-time healthcare applications.

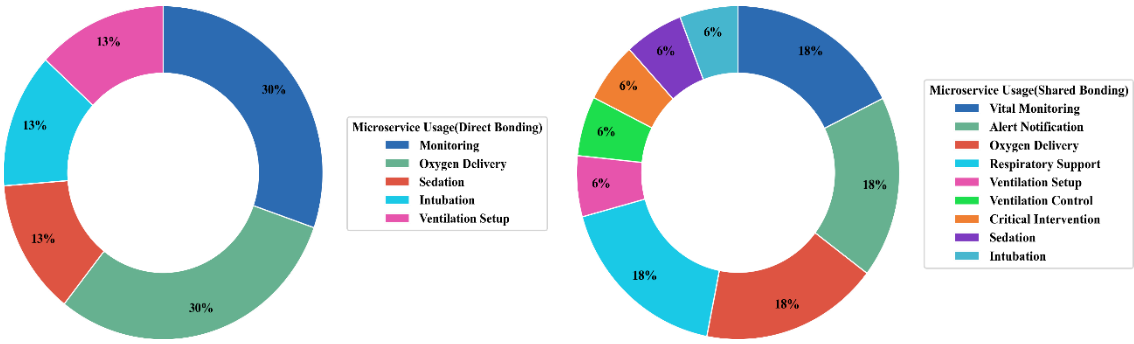

Figure 5. Comparison of microservice usage in direct and shared bonding

5. CONCLUSION

This paper presents a novel self-organizing microservice composition model designed for rapidly evolving healthcare environments that utilize IoT devices. The model has been developed to serve critical medical environments such as NICUs. The model establishes decentralized microservice interaction, enabling autonomous service choreography without needing centralized control. The integration of a deep learning agent enables self-organization and emergent behavior at run time, enabling timely adaptation to patients' needs. The deep learning agent is supported by unsupervised learning through SOM, facilitating the generation of emergent and valid service sequences, strengthening dynamic coordination and adaptation. The model enables predictive service composition through learning, supporting the preloading of services and thereby reducing latency in critical care scenarios. The simulation results are based on authentic clinical data, showcasing real-time learning and service composition. The model has a strong potential for deployment in IoT-enabled critical care settings. The model offers a scalable and responsive solution for real-time healthcare service delivery in an IoT environment. Future research will aim to incorporate adaptive QoS parameters into the service composition process. This enhancement is intended to provide personalized treatment while ensuring resilient and seamless service delivery within a dynamic, IoT-driven healthcare ecosystem.

ACKNOWLEDGEMENT

The authors express their sincere gratitude to the management of SKS Hospital and Post Graduate Medical Institute, Salem, India, for granting permission to observe NICU procedures and for providing access to the required clinical data. The authors would like to specially thank Dr. S. Ramakrishnan, Neonatal Pediatrician, for his expert insights into neonatal care, which greatly informed this research. The authors also acknowledge and thank Dr. Priya Vijay, Family Physician and Chairperson of the Institutional Ethics Committee, for her guidance and support in facilitating the ethical approval process.

FUNDING INFORMATION

Financial aid has not been received from any commercial source or organization.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sharon Poornima	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Ashok Immanuel V	✓	✓			✓	✓		✓		✓	✓	✓		

C : **C**onceptualization	I : **I**nvestigation	Vi : **Vi**sualization
M : **M**ethodology	R : **R**esources	Su : **Su**pervision
So : **So**ftware	D : **D**ata Curation	P : **P**roject administration
Va : **Va**lidation	O : Writing - **O**riginal Draft	Fu : **Fu**nding acquisition
Fo : **Fo**rmal analysis	E : Writing - Review & **E**diting	

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no potential conflict of interest.

INFORMED CONSENT

The data for the neonates used in this research were collected from SKS Hospital and Postgraduate Medical Institute, Salem, India. The Institutional Ethics Committee of the hospital authorized the ethical clearance with a waiver for informed consent due to the retrospective nature of the data.

ETHICAL APPROVAL

All data used in this research were obtained and handled following applicable ethical guidelines obtained from the Institutional Ethics Committee of SKS Hospital and Postgraduate Medical Institute, Salem, India. The data was collected under institutional ethical approval (SKSH/IEC/2024/01/80).

DATA AVAILABILITY

Data availability is not applicable to this paper as the supporting data is not available due to ethical restrictions. The metadata description is available in Zenodo at https://doi.org/10.5281/zenodo.15519118.

REFERENCES

[1] T. Dutta, S. Pramanik, and P. Kumar, "IoT for healthcare industries: a tale of revolution," in *Healthcare Paradigms in the Internet of Things Ecosystem*, Elsevier, 2021, pp. 21–45, doi: 10.1016/B978-0-12-819664-9.00002-8.

[2] D. I. A. Molina, "An algebraic service composition model for the construction of large-scale IoT systems," *Ph.D. thesis*, Department of Computer Science, University of Manchester, Manchester, United Kingdom, 2019.

[3] H. C. Gómez *et al.*, "Evaluating service-oriented and microservice architecture patterns to deploy ehealth applications in cloud computing environment," *Applied Sciences*, vol. 11, no. 10, May 2021, doi: 10.3390/app11104350.

[4] A. Razzaq, "A systematic review on software architectures for IoT systems and future direction to the adoption of microservices architecture," *SN Computer Science*, vol. 1, no. 6, Nov. 2020, doi: 10.1007/s42979-020-00359-w.

[5] M. Hamzei and N. J. Navimipour, "Toward efficient service composition techniques in the internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3774–3787, Oct. 2018, doi: 10.1109/JIOT.2018.2861742.

[6] N. Chen, N. Cardozo, and S. Clarke, "Goal-driven service composition in mobile and pervasive computing," *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 49–62, Jan. 2018, doi: 10.1109/TSC.2016.2533348.

[7] D. Arellanes and K.-K. Lau, "Analysis and classification of service interactions for the scalability of the internet of things," in *2018 IEEE International Congress on Internet of Things (ICIOT)*, IEEE, Jul. 2018, pp. 80–87, doi: 10.1109/ICIOT.2018.00018.

[8] D. Arellanes, "Self-organizing software models for the internet of things: complex software structures that emerge without a central controller," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 7, no. 3, pp. 4–9, Jul. 2021, doi: 10.1109/MSMC.2021.3062822.

[9] S. Poornima and A. Immanuel, “Suitability of self-organizing service composition approach for smart healthcare ecosystem: a study,” *SHS Web of Conferences*, vol. 139, May 2022, doi: 10.1051/shsconf/202213903020.

[10] D. Ye, Q. He, Y. Wang, and Y. Yang, “An agent-based integrated self-evolving service composition approach in networked environments,” *IEEE Transactions on Services Computing*, vol. 12, no. 6, pp. 880–895, 2019, doi: 10.1109/TSC.2016.2631598.

[11] H. B. Mahfoudh, G. D. M. Serugendo, N. Naja, and N. Abdennadher, “Learning-based coordination model for spontaneous self-composition of reliable services in a distributed system,” *International Journal on Software Tools for Technology Transfer*, vol. 22, no. 4, pp. 417–436, Aug. 2020, doi: 10.1007/s10009-020-00557-0.

[12] H. B. Mahfoudh, A. Caselli, and G. D. M. Serugendo, “Learning-based coordination model for on-the-fly self-composing services using semantic matching,” *Journal of Sensor and Actuator Networks*, vol. 10, no. 1, Jan. 2021, doi: 10.3390/jsan10010005.

[13] B. Bekkai, H. Bendjenna, and K. Ilham, “An inspired self-organizing emergent approach for autonomous (IoT) systems,” *Communications in Mathematics and Applications*, vol. 12, no. 3, pp. 755–772, 2021, doi: 10.26713/cma.v12i3.1637.

[14] S. Kalantari, E. Nazemi, and B. Masoumi, “Emergence-based self-advising in strong self-organizing systems: a case study in nasa ants mission,” *Expert Systems with Applications*, vol. 182, Nov. 2021, doi: 10.1016/j.eswa.2021.115187.

[15] A. Bucchiarone, M. D. Sanctis, and N. Bencomo, “Agent-based framework for self-organization of collective and autonomous shuttle fleets,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3631–3643, Jun. 2021, doi: 10.1109/TITS.2020.3021592.

[16] A. Palade and S. Clarke, “Collaborative agent communities for resilient service composition in mobile environments,” *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 876–890, Mar. 2022, doi: 10.1109/TSC.2020.2964753.

[17] S. Hamrioui, J. Lloret, P. Lorenz, and J. J. P. C. Rodrigues, “Cross-layer approach for self-organizing and self-configuring communications within IoT,” *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19489–19500, Oct. 2022, doi: 10.1109/JIOT.2022.3168614.

[18] A. Forestiero and G. Papuzzo, “Recommendation platform in internet of things leveraging on a self-organizing multiagent approach,” *Neural Computing and Applications*, vol. 34, no. 18, pp. 16049–16060, Sep. 2022, doi: 10.1007/s00521-022-07231-3.

[19] R. C. Cardoso and A. Ferrando, “A review of agent-based programming for multi-agent systems,” *Computers*, vol. 10, no. 2, Jan. 2021, doi: 10.3390/computers10020016.

[20] S. Poornima and A. Immanuel, “Self-organizing micro service composition for IoT ecosystem,” in *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, IEEE, Jan. 2025, pp. 273–279, doi: 10.1109/ICMCSI64620.2025.10883497.

[21] N. Singhal, U. Sakthivel, and P. Raj, “Selection mechanism of micro-services orchestration vs. choreography,” *International journal of Web and Semantic Technology*, vol. 10, no. 1, pp. 1–13, Jan. 2019, doi: 10.5121/ijwest.2019.10101.

[22] T. Kohonen, *Self-organizing maps*, Berlin, Heidelberg: Springer, 2001, doi: 10.1007/978-3-642-56927-2.

[23] H. Wang *et al.*, “Adaptive and large-scale service composition based on deep reinforcement learning,” *Knowledge-Based Systems*, vol. 180, pp. 75–90, Sep. 2019, doi: 10.1016/j.knosys.2019.05.020.

[24] M. Dorigo and T. Stützle, “Ant colony optimization: overview and recent advances,” in *Handbook of Metaheuristics*, 2019, pp. 311–351, doi: 10.1007/978-3-319-91086-4_10.

[25] Q. Li, R. Dou, F. Chen, and G. Nan, “A QoS-oriented web service composition approach based on multi-population genetic algorithm for internet of things,” *International Journal of Computational Intelligence Systems*, vol. 7, 2014, doi: 10.1080/18756891.2014.947090.

BIOGRAPHIES OF AUTHORS

Sharon Poornima is currently pursuing a Ph.D. in Computer Science at CHRIST University under the guidance of Dr. V. Ashok Immanuel, with research interests focused on self-organizing microservice composition models for intelligent healthcare systems, particularly within IoT-enabled neonatal and pediatric care. Her work explores the intersection of machine learning, agent-based modeling, and service orchestration to build dynamic, adaptive systems that respond to real-time health data. She is especially interested in integrating QoS constraints, decision-support protocols, and deep learning agents into decentralized microservice architectures to improve responsiveness and reliability in critical healthcare environments. She can be contacted at email: sharon.poornima@res.christuniversity.in.

Ashok Immanuel V earned his Ph.D. in Context-Aware Systems (Ubiquitous Computing) from CHRIST University. He has been a part of the Department of Computer Science since 2005, serving in various academic and administrative roles, including coordinator for information technology (software), admissions coordinator, and currently as the head of the department at the central campus. His research spans across context-aware computing, automata theory, web services, service composition, and ICT in education. A skilled programmer and innovator, he founded the Computer Science Research Centre at CHRIST University in 2022, which drives research in emerging technologies such as artificial intelligence, IoT, extended reality (XR), and drones. He is a frequent speaker at academic and industry forums, including events organized by the Confederation of Indian Industry (CII), and has contributed to modernizing several institutional processes through his software solutions. He can be contacted at email: ashok.immanuel@christuniversity.in.