A Systematic Review on Osmotic Computing

BENAZIR NEHA, Veer Surendra Sai University of Technology, India
SANJAYA KUMAR PANDA, National Institute of Technology, Warangal, India
PRADIP KUMAR SAHU, Veer Surendra Sai University of Technology, India
KSHIRA SAGAR SAHOO, SRM University, Amaravati, AP, India
AMIR H. GANDOMI, University of Technology Sydney, Australia

Osmotic computing in association with related computing paradigms (cloud, fog, and edge) emerges as a promising solution for handling bulk of security-critical as well as latency-sensitive data generated by the digital devices. It is a growing research domain that studies deployment, migration, and optimization of applications in the form of microservices across cloud/edge infrastructure. It presents dynamically tailored microservices in technology-centric environments by exploiting edge and cloud platforms. Osmotic computing promotes digital transformation and furnishes benefits to transportation, smart cities, education, and healthcare. In this article, we present a comprehensive analysis of osmotic computing through a systematic literature review approach. To ensure high-quality review, we conduct an advanced search on numerous digital libraries to extracting related studies. The advanced search strategy identifies 99 studies, from which 29 relevant studies are selected for a thorough review. We present a summary of applications in osmotic computing build on their key features. On the basis of the observations, we outline the research challenges for the applications in this research field. Finally, we discuss the security issues resolved and unresolved in osmotic computing.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Networks** → *Cloud computing*; • **Theory of computation** → Machine learning theory; • **Mathematics of computing** → Evolutionary algorithms;

Additional Key Words and Phrases: Osmotic computing, fog computing, edge computing

ACM Reference format:
Benazir Neha, Sanjaya Kumar Panda, Pradip Kumar Sahu, Kshira Sagar Sahoo, and Amir H. Gandomi. 2022. A Systematic Review on Osmotic Computing. *ACM Trans. Internet Things* 3, 2, Article 9 (February 2022), 30 pages.
https://doi.org/10.1145/3488247

1 INTRODUCTION

Cloud Computing has reached the Slope of Enlightenment in Gartner's Hype Cycle [16]. In fact, the cloud has become resolute underpinnings for next-generation innovations. Even after two

Authors' addresses: B. Neha and P. K. Sahu, Department of Information Technology, Veer Surendra Sai University of Technology, Burla-768018, Odisha, India; emails: neha.benazir13@gmail.com, pksahu_it@vssut.ac.in; S. K. Panda, Department of Computer Science and Engineering, National Institute of Technology, Warangal-506004, Telangana, India; email: sanjayauce@gmail.com; K. S. Sahoo, Department of Computer Science and Engineering, SRM University, Amaravati, AP-522240, India; email: kshirasagar12@gmail.com; A. H. Gandomi (corresponding author), Faculty of Engineering and Information Technology, University of Technology Sydney, Australia; email: gandomi@uts.edu.au.

2577-6207/2022/02-ART9 $15.00
https://doi.org/10.1145/3488247

decades, the cloud continues to evolve with contemporary services that makes it a predominant model for a wide range of solutions. Technological advancements in computing have enabled a multitude of forthcoming internet applications that are compute intensive and latency sensitive in nature. With the outburst of data generated by the digital devices, interaction among data and devices in such applications becomes difficult. The cloud provides unlimited resources and an eclectic mix of services to satiate the requirements of these applications. However, it is strenuous for the geographically centralized cloud datacenters to manage the extensive onslaught of data on their own, which puts an adverse effect on the end user's quality of experience. Although, the cloud provides cost-effective approach to access its services (compute, storage and connectivity), the centralized architecture creates delays and performance complications for data and devices that are at a distance from a centralized cloud or datacenter source. This inefficiency can be addressed with a solution that involves a decentralized approach. Thus, there is a need to push the computing power nearer to data production source. To meet the growing demand for such solutions fog computing and edge computing come into action on par with cloud computing.

Fog computing is the shift away from a centralized, distant cloud and toward local or end-user devices. It is a distributed and hierarchical infrastructure that conciliates between hardware and remote servers to regulate information processing locally [29]. It promotes scalability and handles hefty data for real-time applications with low-latency requirement. Fog computing is a well-known concept that utilizes the devices with computational capacity revolving around us [32]. The devices possessing processing power and storage capacity (like smartphones, switches, routers, etc.) act as Fog devices. Edge computing emphasises processing information locally to enable decision-making instantly [43]. An edge works at the boundary of a network (for example on a laptop not at datacenter). Edge computing and fog computing have led to the disruption of centralized architecture of cloud, pushing cloud's resources nearer to the end users. This transformation is necessary to meet the demands of business opportunities, such as data provisioning in distributed multi-tenant environment, location-based content delivery and other applications in the IoT domain by incorporating cloud solutions [10]. To allow processing and storage abilities on the edge devices without moving data/processing to the centralized cloud, a communication is set up between cloud systems and edge devices. The by product of this set up is edge computing, which also leads to the reduction in volume of data transferred across the edge devices and cloud datacenters. The amalgamation of computing paradigms like cloud and edge is widely conceded in academic and industrial drives, including Cisco, AWS (Greengrass and Snowball Edge), and the OpenFog Consortium.

Osmotic computing is a thriving concept that integrates IoT/cloud/Fog/Edge computing for the dynamic management of IT services (P23) [52]. It is mainly driven by the striking growth in the resource magnitude at the network edge. This paradigm concentrates on techniques to increase IoT servicecs utilizing edge and cloud resources by identifying, designing, and implementing a computing model. Osmotic computing focuses on systematic service allocation using the key principles of osmosis phenomenon in chemistry, and such principles can be used for managing resource and solving data issues in IoT and data science [41]. The principal goal of this new age computing is to balance load and proper utilization of resources between servers without hindering the service deliveries and performances (P25) [19]. Osmotic computing has made its hold in many scenarios, ranging from smart cities to smart healthcare, and connected cars to industry 4.0. It also creates a framework for a dynamic ecosystem where automobiles, wayfarers, and physical infrastructures communicate with one another repeatedly exchanging data on city traffic (P9) [18].

The major contributions of this article are delineated as follows.

- We present a comprehensive analysis of osmotic computing paradigm through a **systematic literature review (SLR)** approach and provide a detailed procedure to conduct a systematic review along with the classification scheme of the fundamental studies.
- We conduct an advanced search on numerous digital libraries to extract the related studies for thorough review and provide a quality assessment on the fundamental studies done in the field.
- We present an explicit and comparative analysis of osmotic computing w.r.t. other computing paradigms.
- We outline the applications and research challenges and discuss the security issues resolved and unresolved in osmotic computing.

To the best of our knowledge, there is no previously published SLR on osmotic computing, as it is still an emerging paradigm. Therefore, we chose to conduct a SLR [1, 4, 25] as an effort to identify current interdisciplinary researches conducted on this new age paradigm. Remainder of the article is structured as follows. The systematic literature review procedure is delineated in Section 2. The review findings are exhibited in Section 3. Section 4 provides detailed analysis, systematic mapping of research questions and fundamental articles, and the limitations of the review. Section 5 draws the attention toward research and security challenges in osmotic computing. The article is concluded in Section 6.

2 THE SYSTEMATIC REVIEW PROCESS

To ensure that this review is of high quality, we meticulously follow reporting guidelines such as the “Preferred Reporting Items for Systematic Reviews and Meta-Analysis: The PRISMA Statement” [30], the “Cochrane Handbook for Systematic Reviews of Interventions” [20], and the “Systematic Reviews in Educational Research: Methodology, Perspective and Applications” [35]. The significant steps to conducting a systematic review are outlined as:

(1) Framing well-defined research questions,
(2) Developing the search strategy,
(3) Defining the selection criteria,
(4) Extracting relevant studies based on selection criteria, and
(5) Assessing quality of the studies.

Each of the steps are briefly explained through out this section.

2.1 Framing the Research Questions

The review question designs a particular structure for each review and impels fundamental decisions about the varieties of the studies to include, where to search for them, how to evaluate their quality, and how to integrate their findings. This SLR aims at procuring the comprehension of osmotic computing with respect to its applications and research directions.

The **Research Questions (RQs)** [26] framed in this SLR acutely focus on the operational aspects of osmotic applications, its research directions, and the maturity level of the relevant studies. The review question intends to shape the selection criteria. To this end, three RQs are raised and shown in Table 1.

2.2 Developing the Search Strategy

The presented systematic review defines search strategies to filter out related studies.

2.2.1 Database Search. We explore various research papers in journals and conference proceedings of high repute that are available in online libraries.

Table 1. Research Questions

RQ_1	What are the distinct aspects of osmotic computing?
RQ_2	How osmotic is comparatively different from other computing paradigms (cloud, fog and edge)?
RQ_3	Which applications followed the significant principle of osmotic computing?
RQ_4	Which key topics are already addressed in osmotic computing?
RQ_5	What is the maturity level of the key topics in osmotic computing?

Table 2. Query String Fed to the Digital Libraries

Keywords	“osmotic computing”	AND	applications
			OR
			usage
			OR
			challenges
			OR
			issues
			OR
			concerns

The various digital libraries searched for this review are enumerated below:

(1) ACM Digital Library (https://dl.acm.org)
(2) IEEE Xplore Digital Library (https://ieeexplore.ieee.org)
(3) ScienceDirect (https://www.sciencedirect.com)
(4) SpringerLink (https://link.springer.com)
(5) Wiley Online Library (https://onlinelibrary.wiley.com)

The sources preferred for the gray literature [44] are as follows:

(1) ArXiv (https://arxiv.org)
(2) Google Scholar (https://scholar.google.com)

2.2.2 *Search Terms.*

The search term is a string that covers all necessary keywords in the RQs along with their synonyms. The array of keywords that build the search string are “osmotic computing,” “applications,” “challenges,” and “advantages.” These keywords are considered as main keywords. We enclose our search term within quotation marks ⟨“ ”⟩ to find exact match of the phrase. Also, we consider the synonyms analogous to previously mentioned keywords for the formation of search string. For instance, issues, concerns, and problems are synonyms for challenges. Additionally, the keywords at variance with word formulation (for instance, string including or excluding non-alphanumeric characters like *hyphen* ⟨−⟩, *colon* ⟨:⟩, and *underscore* ⟨_⟩), are taken into consideration. The query string is framed by combining principal keywords with their synonymous words using logical Boolean operators AND and OR to narrow our search outcome. The query string presented in Table 2 is fed to the web search engine of various online databases and the outcome are found to be distinct from the preferential order of the query terms. For instance, “applications of osmotic computing” and “osmotic computing applications” give same results.

2.2.3 *Search Procedure.*

From the search results of the query string, we extract numerous articles from aforementioned electronic digital databases. However, we found some articles to be irrelevant. To restrict several related articles and filter out irrelevant studies, we use advanced search setting present in various electronic databases. The advanced search setting of each online

Table 3. Quality Assessment Questions

QA_1	Did the article fulfil the purpose of the research?
QA_2	Did the article cover the furnished research question issues?
QA_3	Did the author(s) satisfactorily conduct literature review?
QA_4	Did the article present future research directions?
QA_5	Did the author(s) consider all the important findings?

library considers the metadata that comprise the title of publication, abstract, and keywords/index terms of articles.

2.3 Defining the Selection Criteria

The selection criteria meticulously develop rules that decides which studies can be preferred for the review. The selection process includes studies relevant to the RQs and excludes the irrelevant studies [24]. In this SLR, after initial results are retrieved, irrelevant papers are removed by employing inclusion and exclusion criteria.

2.3.1 Inclusion Criteria. The articles published from November 2016 since osmotic computing emerged as a new paradigm are taken into account. The studies explicitly discussing osmotic computing and its advantages, challenges, and application in various domains published between November 2016 and April 2020 are considered for review.

2.3.2 Exclusion Criteria. The irrelevant articles from the review are eliminated by applying the designed criteria listed below:

- The articles that do not consider osmotic computing as fundamental study.
- The article concerned with other computing models (cloud, edge, and fog).
- The articles written in language different from English and not available in full-text.
- The results showing article summaries, comments, discussions, editorials, prefaces, reviews, reader's letter.

2.4 Assessing the Quality of the Studies

The quality assessment of fundamental research is highly contentious issue in every domain [35]. It is an important phase in a systematic review. Each of the articles is assessed using the **Quality Assessment (QA)** questions [14] presented in Table 3.

In this SLR, we assign scores to the studies based on the satisfaction level of the QA questions. We divide the satisfaction level into three categories such as "satisfactory," "partially satisfactory," and "unsatisfactory," and then assign scores like 1, 0.5, and 0 to them, respectively. The studies satisfying the QA questions are denoted as "satisfactory = 1," partially satisfying studies are denoted as "partially satisfactory = 0.5" and those that are not satisfying the questions are denoted as "unsatisfactory = 0." Subsequently, the qualities of the studies are analyzed on the basis of allocated scores. At the end, cumulative sum of scores of individual study are recorded and the studies with minimum cumulative score of 3 are included in the review (see Appendix A).

The assignment of scores for quality assessment is a difficult task. Consequently, the articles are individually analyzed by the authors of this review in strict adherence to the QA questions mentioned for minimizing the degree of bias.

2.5 Data Extraction

Extraction of data from the relevant studies based on the selection criteria is accomplished and the information regarding the RQs are gathered. The data extracted from each of the fundamental articles are title of the article, data extraction date, publication year, type of article (journal/conference proceedings), publisher name, problem statement, QA score, applications, challenges, and future directions.

3 REVIEW FINDINGS

Studies identified as an outcome of search are examined to ensure fulfilment of the selection criteria. This is done in four phases, namely identification, screening, eligibility, and inclusion [30], owing to detailed analysis of title, abstract, keywords/index terms, and full-text of the articles to ascertain their relevancy to the SLR. The systematic description of the review and selection process of the articles at different phases of SLR are vividly shown in pictorial form in Figure 1. Each phase of the flowchart is described as follows.

3.1 Identification

A total of 99 articles are extracted through the advanced searching on the online database and gray literature. The articles collected at this phase are sent for the screening process in the next phase.

3.2 Screening

3.2.1 Title-based Selection. The title of the articles are chosen on the basis of the selection criteria designed by the authors of this review to find out the relevant studies. The articles with titles reflecting vague description about osmotic computing are excluded from the SLR. On the basis of the title 66 articles are selected.

3.2.2 Duplication Removal. After the title-based selection, the remaining articles are kept in an Excel sheet [4]. The articles are sorted in the order of title, author(s), and source of the article along with the publication year for duplication removal. We found 33 duplicate articles as a result of this screening. After elimination of the duplicate articles, remaining 33 articles are chosen for the next phase.

3.2.3 Abstract-based Selection. Here, we explore the abstracts of the articles and check the keywords/index terms of the chosen articles that cleared the duplication removal phase and extract the related articles with abstracts satisfying the objective of the review. So, we discard 2 articles in the screening phase based on their abstracts and select 31 articles for fundamental studies.

3.3 Eligibility

The selection of studies for a systematic review is a crucial work. Titles and abstracts are insufficient to surmise the relevancy of the articles. In some cases, the title and the content of the article do not match. Therefore, in this phase selection is done based on the full-text to ensure that the studies screened based on title and abstract are eligible to be categorized under relevant studies or not. We search the references of each full-text articles for more relevant studies. After thorough investigation of full-text of 31 articles, we select 29 articles as the ultimate set of studies.

3.4 Inclusion

In this phase, selection is based on QA questions prepared by authors. The quality of each of the selected 29 articles are individually checked by authors against the QA questions mentioned above. The distribution scheme of the fundamental studies obtained from numerous digital roots are shown in Table 4 along with the information related to the extracted studies presented in Table 5. Since osmotic computing was termed in 2016, and it is still in its infancy, most of its research directions discussed in recent years are not accomplished yet. So, we classify the studies on the

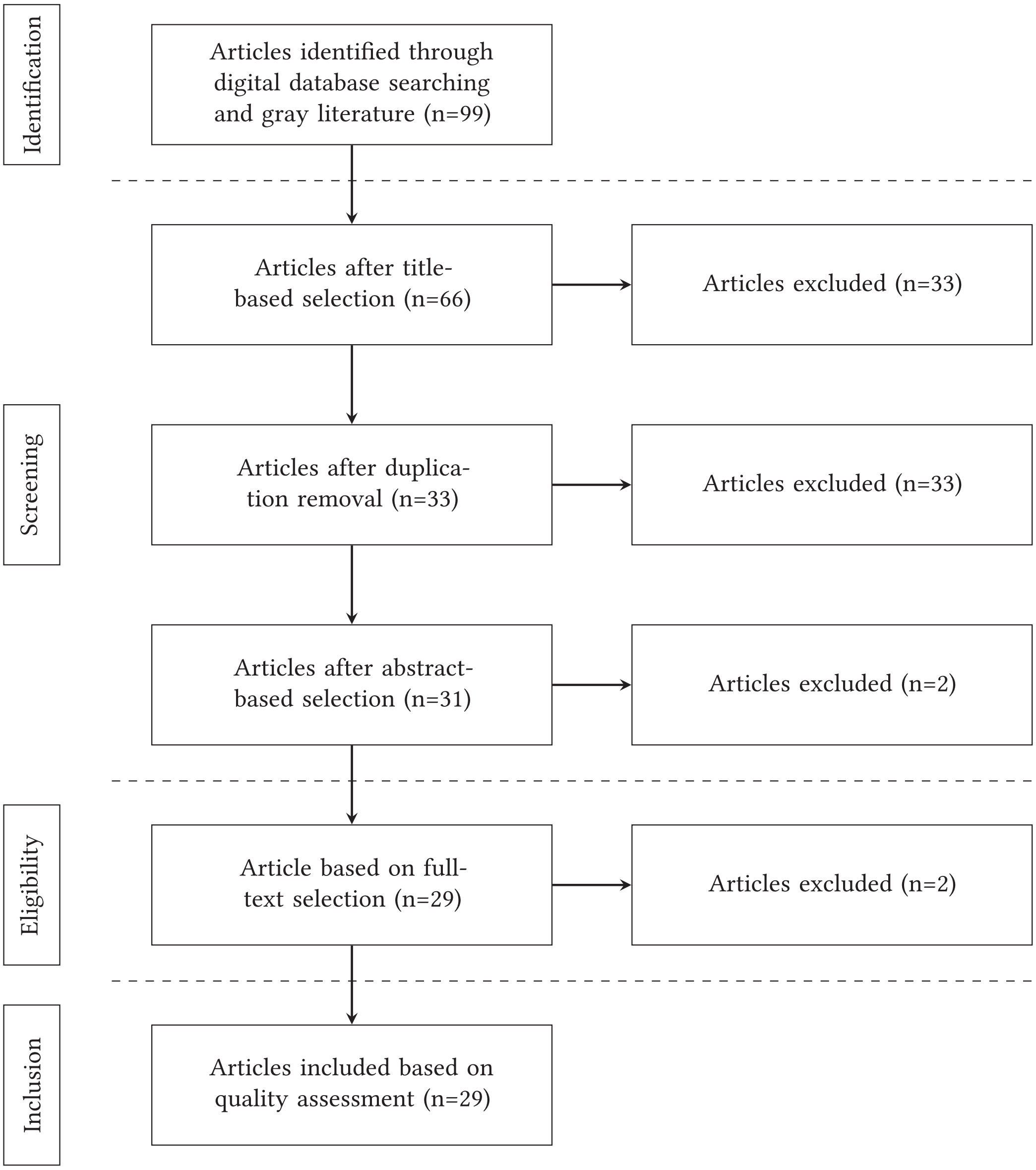
Identification
Articles identified through digital database searching and gray literature (n=99)
Articles after title-based selection (n=66)
Articles excluded (n=33)
Screening
Articles after duplication removal (n=33)
Articles excluded (n=33)
Articles after abstract-based selection (n=31)
Articles excluded (n=2)
Eligibility
Article based on full-text selection (n=29)
Articles excluded (n=2)
Inclusion
Articles included based on quality assessment (n=29)

Fig. 1. Flow of information through different phases of a systematic review.

basis of research maturity into three categories: Initial work, Framework/Model/Middleware, and Usecase/Prototype. The maturity level of the studies are shown in Table 5.

4 DETAILED ANALYSIS

With data becoming the new currency and IoT devices generating capacious data, it becomes challenging to handle the data deluge efficiently. The cloud-centric models, namely Amazon IoT Core and Google cloud Dataflow, typically handle the humongous data produced. However,

Table 4. Distribution Scheme of the Fundamental Studies from Digital Sources

Data Source	Identification	Screening			Eligibility	Inclusion
	Advanced Search	Title-based	Duplication Removal	Abstract-based	Full-Text Based	Quality Assessment Based
ACM Digital Library	6	1	1	1	1	1
IEEE Xplore	25	21	21	21	20	20
ScienceDirect	12	2	2	2	2	2
SpringerLink	12	5	5	4	4	4
Wiley Online	8	1	1	1	1	1
		Gray Literature				
ArXiv	1	1	1	1	1	1
Google Scholar	35	35	2	1	0	0
Total	99	66	33	31	29	29

these models fail to deliver processing power and storage capacity in an effective scale with reduced latency, which makes them inappropriate for the emerging IoT applications. These IoT applications require latency-sensitive solutions, low energy usage, and bandwidth conservation. To respond to the growing demand of IoT applications, the cloud-centric models can be revised to be more adaptable and decentralized (P23) [52]. Thus, cloud and edge resources are exploited to set up a hybrid virtual infrastructure to implement IoT-based computing systems.

Osmotic computing provides a federated environment for such infrastructures to manage cloud and edge datacenters, where distinct providers share their resources for IoT services. Moreover, in such an environment prominence is given to the movement of data and services across cloud datacenters and edge devices. The applications to be deployed on cloud and edge datacenters are decomposed into smaller services called microservices. These microservices are lightweight, resilient, fine-grained, and independently deployable [22]. The advantage of application decomposition into discrete smaller services is to improve modularity, increase cohesion as well as decrease coupling. These factors helps in easy understanding, developing, modifying, and testing of the application. It also facilitates refactoring of applications and continuous delivery.

Microservices are the stand-alone application sub-units/components, applying certain communication protocols for transmitting messages [17]. Software developers deploy various microservices on heterogeneous platforms using explicit interfaces and protocols, without incorporating any particular integration framework [17]. Microservice communication uses a **Representational State Transfer (RESTful)** approach based on HTTP and TCP protocols, XMPP, and **JavaScript Object Notation (JSON)**. This section of review addresses all the RQs discussed before by mapping it with the research findings of the review.

4.1 Osmotic Computing and Its Aspects

Osmotic computing takes analogy from the phenomenon of osmosis in chemistry, where the diffusion of solvent molecules takes place through a partially permeable membrane from a region of higher solvent concentration (lower solute concentration) to a region of lower solvent concentration (higher solute concentration). Similarly, in osmotic computing, microservices are not confined to a specific infrastructure instead they can migrate across cloud datacenters and edge devices through a software-defined membrane called osmotic membrane [53]. A vivid image of this scenario is shown in Figure 2. In the phenomenon of osmosis, we consider various components, each of the components w.r.t osmotic computing are explained below:

- Solute: It represents the static dissolvable component of the solution that cannot pass across the semi-permeable membrane. In osmotic computing, solute generally refers to various computational properties like processing time, load, power, energy, and so on. The selection of properties may vary according to the application's need (P6) [47].

Table 5. Details of Extracted Fundamental Studies

PID	Author(s)	Year	Objective	Approach	Research Maturity	QA Score
P1	Villari et al. [54]	2016	To introduce osmotic computing as a new paradigm.	Theoretical	Initial work	4
P2	Nardelli et al. [33]	2017	To design an osmotic flow model for IoT workflow applications.	Data analysis programming	Model	3.5
P3	Morshed et al. [31]	2017	To develop holistic distributed deep learning model for heterogeneous computing models.	Deep Learning	Initial work	3
P4	Okafor et al. [36]	2017	To evaluate virtualized osmotic cloud network.	Discrete event simulation heuristic	Usecase/prototype	4.5
P5	Sharma et al. [48]	2017	To design a pervasive trust management framework.	Ant Colony and Artificial Bee Colony Optimization	Usecase/Prototype	4.5
P6	Sharma et al. [47]	2017	To develop a fitness-based osmosis algorithm for managing service heterogeneity.	Fitness function	Initial work	4
P7	Oyekanlu, [37]	2018	To generate coiflet wavelet for C28xDSP at IIoT network.	Deep Learning	Usecase/Prototype	3.5
P8	Villari et al. [55]	2018	To exploit blockchain technology in software-defined membrane.	Blockchain	Usecase/Prototype	3
P9	Filocamo et al. [18]	2018	To develop intersection management algorithm for autonomous vehicles in self adapting city.	V2V	Initial work	3
P10	Oyekanlu, [38]	2018	To generate osmotic collaborative Gaussian mixture model to optimize application integration in resource-constrained IoT network.	Gaussian Mixture Models	Model	4
P11	Longo et al. [27]	2018	To develop an efficient urban population monitoring osmotic middleware based on mobile crowd sensing.	Machine Learning	Middleware	4.5
P12	Souza et al. [51]	2018	To build effective load balanced osmotic framework for smart city applications.	Live migration	Framework	5
P13	Pachecho et al. [39]	2018	To develop osmotic smart lighting prototype for person recognition in smart classroom.	Deep Learning	Usecase/Prototype	3.5
P14	Rausch et al. [42]	2018	To develop osmotic message-oriented middleware to accelerate communication between devices in IoT environment.	MQTT Protocol	Middleware	3
P15	Souza et al. [50]	2018	To develop osmotic monitoring system for microservices deployed across cloud and edge layers.	Cross layer monitoring	Usecase/Prototype	4.5
P16	Buzachis et al. [5]	2018	To develop osmotic-based health information technology (HIT) system for ISASI-CNR (ME).	Neural Network	Usecase/Prototype	5
P17	Buzachis et al. [6]	2018	To develop an osmotic flow model for game-based cognitive rehabilitation usecase.	Virtual reality system	Usecase/Prototype	4.5
P18	Buzachis et al. [9]	2018	To propose MEL orchestration approach in osmotic environment using blockchain-based SDMem.	Blockchain	Initial work	4.5
P19	Carnevale et al. [11]	2018	To design multi-tenant osmotic-based architecture for MEL orchestration.	Artificial Intelligence	Usecase/Prototype	3.5
P20	Buzachis et al. [8]	2018	To analyse microservice deployment and connectivity issues leveraging different network overlays.	Container technology	Initial work	4.5
P21	Carnevale et al. [13]	2018	To develop osmotic-based Big Data hospital information system (HIS) for IRCCS-ME.	Virtualization	Initial work	4
P22	Sharma et al. [46]	2018	To design an osmotic-based framework for exploring 802.21a-2012 MIH in IoT-Fog ecosystem.	Osmotic absorption key	Framework	3.5
P23	Villari et al. [52]	2018	To introduce concept of microelements (MELs) to support development of future internet services.	Theoretical	Initial work	3
P24	Kaur et al. [23]	2019	To develop an osmotic-based framework to guarantee energy-aware dynamic management of resources.	Hyper Heuristics	Framework	3.5
P25	Gamal et al. [19]	2019	To design an osmotic-based hybrid meta-heuristics for cloud computing by exploiting bio-inspired load balancing algorithms.	Ant Colony and Artificial Bee Colony Optimization	Framework	5
P26	Carnevale et al. [12]	2019	To develop an osmotic-based architecture incorporating a distributed multi-agent system.	Artificial Intelligence	Initial work	5
P27	Buzachis et al. [7]	2019	To design FaaS-based osmotic flow model for smart city video surveillance.	Serverless computing	Usecase/Prototype	4.5
P28	Longo et al. [28]	2019	To design and develop an osmotic-based middleware for urban pollution monitoring in a hybrid network.	Machine Learning	Usecase/Prototype	5
P29	Sharma et al. [45]	2020	To develop an osmotic-based framework to ensure service migration and efficient resource scheduling in an augmented reality system.	Optimization functions	Framework	4

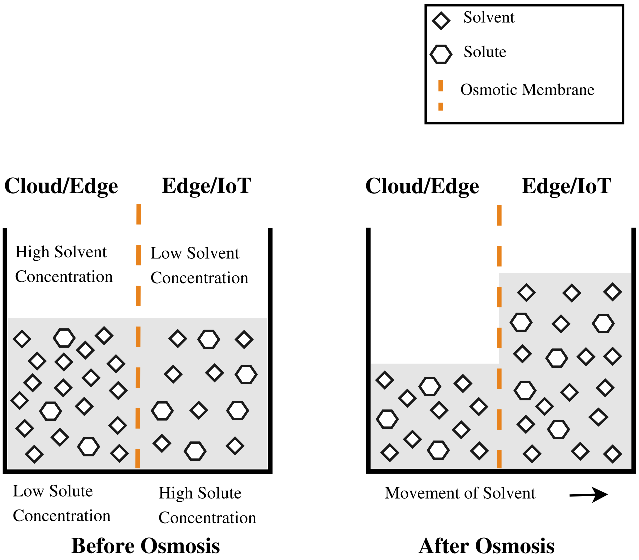

Fig. 2. Osmotic computing concept.

- Solvent: This is a component that dissolves solute resulting in a solution. In osmotic computing, solvent indicates the applications, services as well as microservices. This component is allowed to migrate across the semi-permeable membrane.
- Solution: The solute and solvent are combined together to form the solution. In osmotic computing, the entire federated environment comprising microservices, applications, resources, and users for the solution.
- Semi-permeable membrane: It plays the role of a decision-making system that allows movement of solvent across the solution to maintain balanced concentration w.r.t. solute. In osmotic computing, it is also known as osmotic membrane that is a **software-defined membrane (SDMem)** capable of allowing microservices to pass through it.
- Concentration: It is the ratio of solute to solvent.

Definition. Villari et al. (P1) [54] concisely defined osmotic computing as *a new paradigm that is driven by the significant increase in resource capacity/capability at the network edge, along with support for data transfer protocols that enable such resources to interact more seamlessly with datacenter-based services.*

Osmotic computing enables tunable configuration of resources based on application requirements and resource availability that makes it distinguishable from the chemical osmotic process (P28) [28]. This distinct feature helps in determining the direction of microservice movement, i.e., from cloud to edge and in reverse. This dynamic bidirectional flow of microservices makes osmotic computing stand out the simple elastic behaviour of resource deployment.

Various aspects of osmotic computing can be rendered as follows:

(1) Highly distributed and federated environment for automatic deployment of microservices.
(2) Seamless collocation of cloud and edge for service migration.
(3) Tunable configuration of resource utilization based on application requirements and resource availability.

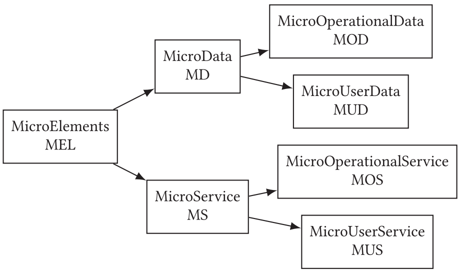

Fig. 3. Hierarchical structure of MEL.

(4) Controlling and managing service heterogeneity.
(5) Reduction in latency and energy consumption.

4.1.1 Osmotic Elements. The integration of cloud, edge, and IoT results in movement of services, resources, and data across heterogeneous infrastructures. An IoT application in a federated and distributed environment is represented as a graph of **microelements (MELs)** (P19) [11]. MELs are the software and data abstractions that are bifurcated into components, namely **MicroService (MS)** and **MicroData (MD)**. The hierarchical structure formed from these abstracted components is shown in Figure 3. MS implement distinct functionalities that are deployed and migrated across heterogeneous platforms. MD characterize bidirectional flow of domain-specific data (passive and active data) across IoT devices. Passive MD allow devices to read and update them, whereas active MD are allowed to be queried (MongoDB). The leaf of the hierarchy shows further categorization of MS and MD based on user (U) and operational (O) access. MS is branched into MUS (e.g., User Application) and MOS (e.g., Operating System), MD is branched into MUD (e.g., User Data) and MOD (e.g., MS Configuration). The flow of MELs across different infrastructures are regulated by osmotic membrane. This osmotic membrane is an SDMem that allows collecting and filtering microservices on the basis of their salient features allowing them to migrate, conditional upon the constraints present in the membrane, obscuring the systems from each other (P8) [55]. This isolation ensures security and privacy, restricting any intrusion on sub-parts of the complete system. MELs are hosted in cloud, edge, and IoT infrastructures and can migrate across them elastically as shown in Figure 4.

The osmotic computing paradigm concentrates on discerning the different microservices types that are appropriate to be executed at the edge/IoT instead of cloud datacenter and vice versa. However, the load of data-offloading across multiple datacenters elevates energy consumption. To mitigate this issue and ensure dynamic management of resources, a reinforced decision making framework namely **energy-aware osmotic computing (En-OsCo)** is designed (P24) [23].

The cloud exploits the hypervisor virtualization to manage virtual machines, and the virtual machines exploit the container virtualization to run containers and the containers in turn run microservices. The edge infrastructure manages several edge devices that can run containers. Container-based virtualization technologies (e.g., Linux Container, Docker Container, Google Container, and Amazon Compute cloud Container) are used for deployment of MELs on cloud and edge datacenters, whereas Python-VM, Lua-VM, and Javascript on IoT infrastructure surface as a light-weighted substitute to hypervisor-based technologies (e.g., Xen and Hyper-V) used in cloud (P23) [52]. Containers lead to reduction of deployment overhead by allowing explicit software components (like a database server) to be enclosed. These container-based technologies strictly adhere to the physical host's abilities and characterizes microservices deployment on resource-restricted

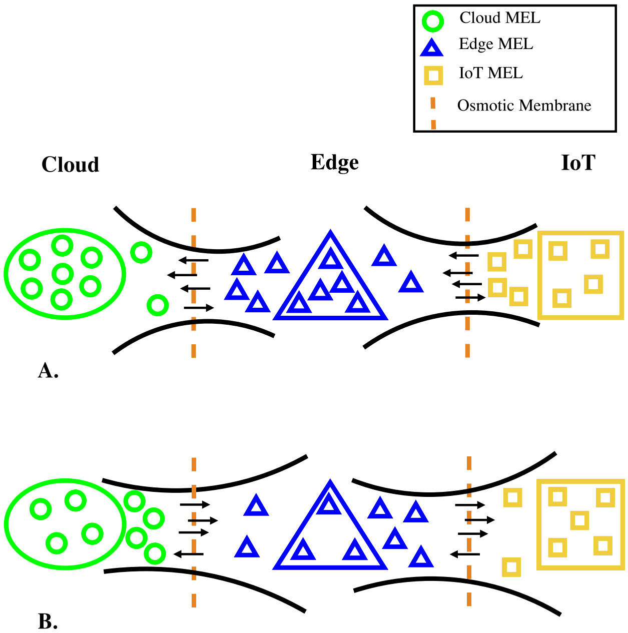

Fig. 4. Migration of MELs across cloud, edge, and IoT: (A) Migration of MELs from Cloud to Edge and from Edge to IoT. (B) Migration of MELs from Edge to Cloud and from IoT to Edge.

smart devices available at the edge of the network like gateways (Raspberry Pi and Arduino), network switches (HP OpenFlow), and software-defined network solutions (Cisco IOx) [2].

Osmotic computing enables automatic adaptation of microservices to the deployment sites based on deployment locations. In this scenario, a decision maker maps microservices to the related location and the decision is affected by the constrained application as well as the infrastructure provider. To address heterogeneity in microservices and their runtime deployment, an orchestrator like kubernetes is used. Kubernetes[1] is an open source container-orchestration system that facilitates the work of deploying and removing microservices on remote fixed devices in association with container tools like Docker.

4.1.2 Osmotic Ecosystem. In the federated ecosystem, osmotic computing spans across cloud, edge, and IoT infrastructure layers. In such an environment, osmotic computing appear as an application-agnostic framework and offers runtime environment to user applications in a distributed manner. Thus, the prime types of microservices, this application-agnostic framework should set up and deploy into different infrastructures (cloud, edge, and IoT) are general-purpose microservices, network-management microservices, and security-management microservices. The general purpose microservices are orchestrated for a particular application goal, network management microservices are deployed to organize virtual networks among microservices (P20) [8] and security management microservices reinforce development of cross-platform security-enabled microservices (P18) [9].

In this ecosystem, Cloud resides on Layer 1, Edge is available on Layer 2 and IoT devices are placed in Layer 3 closer to end users. Layer 1 hosts conventional applications (computational and data-storage solutions, data analytic platforms) controlled by cloud providers. At this layer,

[1] kubernetes, https://kubernetes.io/.

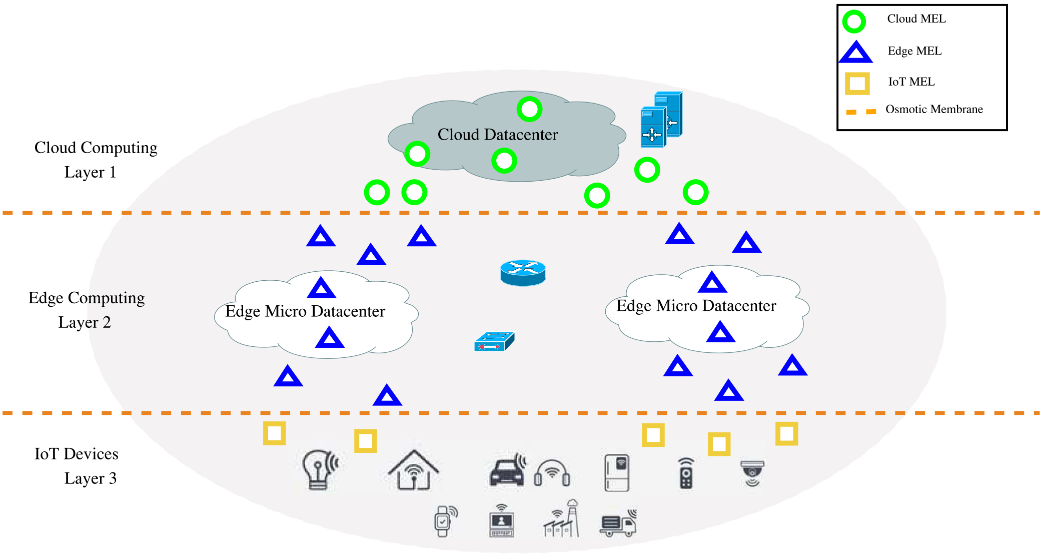

Fig. 5. Osmotic ecosystem.

microservices are formed in accordance with the users' requirements. Here, the operations are based on pre-defined objectives (such as throughput, cost, and response time) between client and cloud datacenter. In Layer 2, MELs are deployed on edge devices that implement container-based virtualization technologies (Gateways: Raspberry Pi and Arduino). At this layer, computational operations (like filtering, aggregation) are done on local data through micro-datacenters available in close proximity of end users. These local data are acquired from Layer 3 and operate in Layer 2 before being forwarded to Layer 1 for subsequent analysis. In Layer 3, IoT devices set up communication based on well-defined protocols like **Constrained Application Protocol (CoAP)**, backed by a RESTful interface, proficient operating systems (Contiki and RiotOS) plus network functions like **Open Virtual Network (OVN)**. OVN is an open source network virtualization abstraction introduced by Open vSwitch (currently belongs to VMware) that can help in establishing MEL communication in a network and preserving device-centrical, data-centrical as well as network-centrical security.

The applications hosted on cloud (Layer 1) correlate with Edge (Layer 2) and IoT (Layer 3) present in the ecosystem with the help of an orchestrator. The orchestrator configures movement of microservices across different infrastructure layers in response to the user runtime requirements, QoS, and security concerns. The osmotic membrane along with the orchestrator takes the decision about the movement of microservices to maintain appropriate balance. In case of a streaming application, edge (Layer 2) can filter out and accumulate traffic flow from source devices (Layer 3), where as queries with hefty processing or archival outcomes can be passed to the cloud (Layer 1). A pictorial representation of the osmotic ecosystem distinguishing each layer is shown in Figure 5.

4.2 Comparative Analysis of Osmotic Computing with Other Computing Paradigms

Osmotic computing is flourishing as a neoteric paradigm that integrates several distributed computing models to ensure service migration and service heterogeneity. It aims to support the seamless deployment of application and services across the heterogeneous infrastructures (IoT, Edge, and Cloud) according to current resource availability and application needs. In this section, we compare osmotic computing with different distributed computing models (cloud, fog, edge, mobile-edge, and cloud offloading).

Cloud computing brought a substantial change in the IT industry by rendering computing as a utility. It provides availability of infinite computing resources on-demand in pay-per-use basis. On the contrary, osmotic computing advances the cloud computing resources to the devices at the network edge to deliver seamless services by deploying MELs. Additionally, cloud provides elasticity of resources and on-demand scalability of workloads within the datacenters. Osmotic computing exhibits the elastic behaviour of resource deployment at the network edge as well as across other layers through bidirectional migration of microservices.

Fog computing complements and outstretches cloud paradigm to the network edge [3]. It performs fast analysis and data processing for time-sensitive applications. However, data that do not require time-sensitive processing are analysed, stored, and processed in the cloud. Osmotic computing takes part in the migration of services to reduce latency and support real-time responsiveness by abstraction of MELs, continuous monitoring, and feedback-driven orchestration. Fog computing seamlessly consolidates edge devices and cloud resources to handle multiple application at the edge. Osmotic computing also creates an ecosystem to integrate cloud and edge, but instead of running the entire application it decomposes the applications into microservices and deploy them across cloud and edge datacenters.

Edge computing and fog computing are used inter-changeably across the literature of computing paradigms to refer to the processing of data/traffic at/near the periphery of network away from the centralized architecture [49]. Both the paradigms bring cloud processing capabilities closer to where the data are originated. However, edge computing is a step ahead from fog computing; it embeds data processing capabilities directly into the end devices circling around us. **Mobile edge computing (MEC)** extends edge computing by rendering computation and storage closer to mobile devices. It also provides an ecosystem to migrate compute intensive tasks to nearby edge servers [21]. In a wide scenario, osmotic computing moves the edge resources to the cloud datacenters and cloud resources to the edge micro datacenters leading to dynamic management of resources through osmotic orchestrator.

MEC allows splitting of applications into smaller tasks analogous to osmotic computing. However, it faces difficulties while distributing tasks among the edge and cloud. In contrast, osmotic computing efficiently distributes tasks across its ecosystem. MEC resides between cloud and mobile devices and complies with cloud to amplify performance of the end-devices. However, osmotic computing resides as a layer between each infrastructure present in its ecosystem to improve computational offloading, data storage, and processing. With increased usage of mobile devices and demand for execution of compute intensive applications on the mobile devices, mobile cloud computing extends the capabilities of mobile devices using offloading techniques. Cloud offloading [15] is one of the computational offloading techniques that provides the solution to enhance the abilities of mobile devices by migrating resource-intensive tasks to servers with high computational resources. Osmotic computing works in line with the concept of computational offloading and tackles certain limitations of mobile devices like limited battery life, processing ability, and storage capacity. Osmotic computing presents an approach that combines mobile offloading (computation offloading from IoT/edge devices to cloud datacenters) and datacenter offloading (computation offloading from cloud datacenters to IoT/edge devices) [48]. This approach helps in minimizing latency and reducing volume of data to be migrated over the network. A Comparative analysis reflecting different characteristics of cloud, edge, fog, mobile-edge along with osmotic computing is shown in Table 6. From this analysis, we can say that osmotic computing excel as a new generation computing paradigm.

Table 6. A Comparative Analysis of Osmotic Computing with Other Computing Paradigms

Characteristics	Cloud Computing	Fog Computing	Edge Computing	Mobile Edge Computing	Osmotic Computing
Architecture	Centralized	Distributed	Distributed	Distributed	De-centralized/Distributed
Proximity	Away from end users	Relatively close to end users	Close to end users	Close to mobile users	Depends on computation
Application Type	Latency-tolerant	Latency-sensitive	Latency-sensitive	Delay-sensitive	Real-time
Data Processing	Remote Datacenters	Edge of the Network	Edge devices	Edge Servers	Edge/Cloud
Hardware	Few large Datacenters	Several small servers	Edge devices	Small Datacenters	Micro Datacenters
Latency	High	Low	Low	Low	Low
Scalability	High	Moderate	Low	—	High
Computing Capabilities	High	Moderate	Moderate	Moderate	High
Storage	High	Moderate	Low	Low	High
Resource Availability	High	Moderate	Moderate	Moderate	High
Responsiveness	Low	High	High	Moderate	High
Data Traffic	High	Low	Low	Low	High
Bandwidth Conservation	Low	High	High	High	High
Fault-tolerance	Low	High	High	High	High
Security	Low	High	High	High	High
Power Consumption	High	Low	Low	High	Low

4.3 Application of Osmotic Computing

Currently, half of the global enterprises are using various forms of cloud computing services. Several large companies like Apple and Netflix use cloud-based applications to triumph in the new age market. However, exploiting cloud services for latency-sensitive applications can be exceptionally difficult considering the fact that the centralized management of resources can cause delays. Hence, new computing paradigms such as fog and edge are developed. Moreover, the coexistence of these computing models for a specific application can be cumbersome in terms of service management and interoperability.

Osmotic computing can overcome these issues by combining services offered across edge devices and cloud datacenters and allowing service migration from datacenter to a device closer to the user. The advancement in development of infrastructure and intelligence have led to many usage plots ranging from smart homes, mhealth, logistics and supply chain, smart transport, and connected cars to wearable devices. However, the efficacy of these technologies needs coordination among various layers, inducing future research challenges. Table 7 shows the summary of substantial work done in osmotic computing platform along with their features and research challenges related to them.

4.3.1 Healthcare. It is one of the most growing sectors that facilitates health improvement and healthcare services. The amalgamation of the wearable devices, digital technologies, and ubiquitous computing have dramatically revolutionized the healthcare solutions. This merging allows more accurate electronic health records, remote health monitoring, and medical diagnostics, which gives patients increased access to information and control over their health through apps.

Healthcare has shifted from traditional storage and processing approaches to advanced technological trends. The traditional **healthcare information technology (HIT)**/**healthcare information system (HIS)** has various limitations, such as data management, fault-tolerance and scalability. In certain situations, such as pandemics, mass agitations, lock downs, and catastrophes, a healthcare infrastructure must be well equipped, digitalized, and readily available with modern tools and technologies to handle this. In this context, various healthcare need to shift from cloud-centric models and adopt osmotic computing models to meet the emerging healthcare application needs. In this line, modernization of the healthcare IT infrastructure is a next big step.

Carnevale et al. (P21) [13] discussed an approach to deploy a HIS for IRCCS-ME (a healthcare and research center) to store and process Big e-health Data collected from various medical devices. They presented numerous reasons for migration to take place from conventional HIS to osmotic-based HIS solution to improve data and service management. They also proposed an idea to create a robust infrastructure by incorporating the principles of osmotic computing and executing the hospital operations as microservices. An identity and access management service was proposed for management of users' authentication and authorization for their entire healthcare system using lightweight directory access protocol. Motivated from this work, Buzachis et al. (P16) [5] proposed a design to modernize the HIT system of ISASI-CNR (ME) based on osmotic computing paradigm. They presented two microservices-based usecases Dyslexia and Ataxia to show the complexities of the patients suffering from these disabilities. The Usescases incorporate bifrost infrastructure in their architecture. Under Dyslexia usecase two projects SMART@PP and DyslexiaOnline were created to detect the disorder at the beginning stage and to create an digital platform for remote evaluation as well as recovery of reading problems of children/adults, respectively. Ataxia usecase was developed to evaluate patients experiencing progressive ataxia, non-progressive ataxia, and control subjects.

With IoT becoming an inseparable part of the modern healthcare industry, **internet of medical things (IoMT)** is sweeping in as a revolution. IoMT helps in reducing inessential hospital visits and the load on healthcare systems by connecting patients to the health professionals and exchanging medical data on to a secure network. To provide hassle-free facilities IoMT seeks ultra-high reliability along with low latency. Buzachis et al. (P17) [6] investigated the osmotic computing principle to propose an osmotic workflow model employed to a real gamified cognitive rehabilitation usecase. The usecase developed a custom-built virtual reality system allowing the patients to conduct rehabilitation therapies (physical and cognitive) using a microsoft kinect-based UI. The kinect sensors were connected to a Raspberry Pi model. Carnevale et al. (P26) [12] presented a distributed multi-agent system-based osmotic architecture. A **Body Area Network (BAN)** scenario was demonstrated for orchestrating microservices across Cloud/Edge/IoT. A logical architecture of an osmotic agent was presented to show the dynamic migration of microservices. The response in health status alarm was considered as key performance indicator.

The advances in healthcare systems is attracting various technologies toward it. Deep learning is one of the technologies that is compelled by this domain. It helps in predicting and suggesting medical diagnostics to improve patient's health. Morshed et al. (P3) [31] investigated and orchestrated deep learning technology across Cloud/Edge/Mobile-Edge environment. The research challenges associated with the development of a class of **holistic distributed deep learning (HDDL)** algorithms for the heterogeneous infrastructure were analyzed. A vision of HDDL was presented for deploying and orchestrating it as mobile-edge (e.g., Fitbit), edge (Cisco IoX routers and OpenFlow-based SDN), and cloud. A biomedical application-telemedicine was considered to demonstrate the vision of HDDL for a person diagnosed with muscle sympathetic nerve activity that may lead to heart attack.

4.3.2 Monitoring System. The integration of heterogeneous infrastructures introduces certain complexities in deploying microservices. The dynamic migration of microservices across datacenters, micro datacenters, routers, and gateways makes it difficult to detect faults, which leads to QoS degradation. There are various monitoring frameworks/techniques for cloud and edge such as Amazon CloudWatch, Heapster, and so on, to monitor resources (CPU, memory, and storage) and network usage. However, very few monitoring techniques/tools/frameworks are developed that can monitor QoS parameters, workloads, and so on, at microservices level across cloud/edge. Researchers have come up with monitoring techniques implementing osmotic computing principles across multiple layers.

Longo et al. (P11) [27] designed and developed a middleware for pollution monitoring in urban areas using osmotic computing paradigm. Apollon platform was used to implement and experiment the new methods and tools for analyzing environmental factors w.r.t. pollution, UV, and atmospheric monitoring. The Apollon middleware used low-cost sensors to gather data from

Table 7. Application of Osmotic Computing

PID	Application	Features	Research Challenges
P28	Osmotic computing infrastructure for urban pollution monitoring [28]	• It presents an osmotic-enriched middleware architecture to manage sources in Apollon platform developed under Apollon project framework.[2] • The urban pollution monitoring system exploits numerous advantages of osmotic computing for creation of an abstract layer across the smart devices and cloud that refines the metadata for progressive processing and load distribution.	• To develop standard data provenance and qualification algorithms taking edge devices into consideration.
P26	BAN [12]	• It is an interdisciplinary domain that enables regular health monitoring and logging vital signs of patients via internet. It represents systems where interaction is intensely within the proximity of a human body [34]. • It tackles the intensification of data and complex cases, by implementing osmotic computing concept in BAN to obtain reliable health monitoring outcomes through distributed multi-agent system.	• To validate osmotic computing architecture through an implementation along with a comparison with the existing architecture.
P24	En-OsCo [23]	• It aims at allowing service migration across osmotic layers. • It diligently monitors edge resources and transmits services to edge/cloud layout incorporating Hyper-Heuristics. • It helps in reducing the energy usage by data centers while fulfilling QoS requirements.	• To introduce energy-aware dynamic management of resources in social multi-media applications.
P12	Osmotic Execution Framework [51]	• It develops a framework for smart city application by incorporating osmotic computing concept in a distributed ecosystem to ensure dynamic load balancing. • This framework uses containerization that allows migration of microservices across edge and cloud for efficient resource utilization.	• To estimate the count of microservices to be deployed at the edge to upgrade the functioning capabilities of the smart city application.

(Continued)

[2] Apollon Project, http://apollon-project.it/.

Table 7. Continued

PID	Application	Features	Research Challenges
P17	Cognitive Rehabilitation flow model [6]	• This is a scientific work in the domain of e-health that develops a flow model in a closed-loop game-based rehabilitation use case by implementing the fundamentals of osmotic computing. • This model provides a custom-tailored game-based virtual reality system that assists patient to complete rehabilitation therapies using a real-time user interface.	• To implement artificial intelligence (AI) techniques analyzing the patient's rehabilitation sessions.
P14	Osmotic MOM [42]	• This work aims at developing a message-oriented middleware to optimize end-to-end delays for users and managing on-demand elastic diffusion of brokers and users to the edge. • It ensures seamless and transparent control mechanisms to optimize responsiveness of clients. It allows customized configuration and development of brokers through proximity detection.	• To guarantee message delivery while allowing broker mobility. • To scale proximity monitoring of brokers.
P15	Osmotic Monitoring [50]	• This work develops a model with advanced monitoring solution for IoT applications like smart parking system. This model aims at analyzing the QoS metrics of any application by examining containerized microservices present in the cloud/edge platforms. • This monitoring model pinpoints variations in the resources across cloud and edge at the application level, microservice level and infrastructure level.	• To expand the model for device monitoring by extending analysis and execution of workload.
P13	Smart Classroom [39]	• It presents a smart lighting control prototype that integrates vision monitoring system and an osmotic-IoT architecture to identify objects, people and handwritten texts through deep learning models. • The goal is to find the performance excellency of the osmotic layer (edge, fog and cloud) while executing a particular deep learning model.	• To implement a faster and upgraded network protocol. • To explore new candidates for implementing fog IoT microservers.

(Continued)

Table 7. Continued

PID	Application	Features	Research Challenges
P16	HIT Systems of ISASI-CNR (ME) [5]	• The purpose of this work is to explore the infrastructure of ISASI-CNR and digitize the clinical routine. In these HIT systems, the conventional systems are relocated to revolutionary infrastructures incorporating osmotic computing principles. • Ataxia and Dyslexia projects are the two real use cases that employs microservices-based approach and are incorporated in patients' homes as well as healthcare infrastructures in collaboration with ISASI-CNR.	• To develop more real use cases implementing the Osmotic-based HIT system.
P2	Osmotic Flow [33]	• It is an integration of osmotic computing concepts with IoT workflow to strengthen existing IoT workflow solutions for executing data analytics in cloud-edge environment. • It is a flexible programming model that focuses on processing continuous data traffic on cloud/edge datacenters.	• To develop autonomous workload coordination protocol to ensure load balancing and resource utilization in cloud/edge datacenters.
P3	Deep Osmosis [31]	• In collaboration with holistic distributed deep learning technology, osmotic computing ascertains resource and data aware solutions for biomedical applications as well as other applications in such domain to enhance compatibility between multiple deep learning models.	• To develop an API to support seamless access to heterogeneous data mapped to osmotic computing environment.

heterogeneous sources (edge, IoT, and **mobile crowd sensing (MCS)**) and developed a data layer to integrate them. They applied osmotic computing principles to the middleware for controlling, abstracting, and opportunistically filtering data as per the requests from the cloud. A microservices orchestration system was developed for the configuration, deployment, and management of microservices. A smart system build on machine learning was applied in the microservices configurator to make the management of microservices autonomous. In case of unavailability of resources at edge/cloud, a message broker (Apache Kafka) was introduced in the middelware to storing messages temporarily in a queued manner and resume services when the resources are available. They validated the middleware by carrying out unit, integration, and load tests on Agento, kubernetes, and kafka filtering applications. Longo et al. (P28) [28] designed and developed a prototype to explicitly address the connection between conventional network monitors, IoT-fixed devices (Laptops), and mobile devices (Smartphones). A middleware was developed for the interaction amongst cloud, edge, IoT, as well as MCS layers with message brokering facilities. This acts as an interface for heterogeneous data coming from the smartphone users and edge devices. The prototype was designed with strict adherence to data provenance, data quality, and privacy of users participated in crowd sensing. JSON packets were used to send data enveloped with kafka messages. At the cloud side, Apache Spark was adopted as big data processing platform for complex aggregation and computation clubbed with NO-SQL database (MongoDB). The system prototype included a mobile app (City SenseScape) to gathering as well as processing various parameters of acoustic pollution (particulate matters, temperature, relative humidity, etc.) as per MCS needs. They also proposed an osmotic membrane prototype wrapped with edge devices that aims at filtering out hostile packets along with sending messages to the accurate broker. They validated their system with various technical tests.

Souza et al. (P15) [50] presented an IoT application cross-layer monitoring system in an osmotic computing ecosystem. A smart-parking application was considered for performance and experimental evaluation. The osmotic monitoring system holistically identified variations in QoS parameters (CPU, memory, and network latency) of deployed microservices in cloud/edge layers using containers.

4.3.3 Blockchain. Osmotic computing facilitates dynamic movement of MELs across heterogeneous infrastructures in a federated environment. Moreover, the flow of MELs in this ecosystem is regulated by an osmotic membrane (SDMem). The SDMem allow swift migration of microservices adhering to the constraints of the membrane and privacy of each layers present in the ecosystem. This isolation helps in ensuring security, privacy policies, and selective sharing of resources. However, the complex nature of the federated environment due to collaboration between different infrastructures with different security requirements impel the need of a strong administration and efficient security policies. In line with this, researchers are leveraging blockchain facilities for MEL orchestration and improving access control system. Blockchain is a record-keeping technology that solves the challenges related to data management. It provides transparency, resilience and integrity.

Buzachis et al. (P18) [9] proposed a secure MEL orchestration through SDMem by leveraging blockchain in an osmotic environment. This work addressed security problems related to data, microservices as well as shared resources by the member organisations in an integrated ecosystem. A permission-based private blockchain was adopted for storing shared data and records. They proposed the adoption of Hyper ledger Fabric technology to develop an access control system. Villari et al. (P8) [55] explored private blockchain technologies and presented a prototype for MEL orchestration through SDMem. They used various levels of encryption in private/federated/hybrid models of blockchain to ensure confidentiality and integrity of MELs. A comprehensive architecture was designed using private blockchain and JSON object signing and encryption for MEL management. The private blockchain were used to guarantee ownership and integrity of membranes.

4.3.4 IoT Workflow and Industrial IoT. With the ascending IoT devices around the globe and its increasing applications in domains ranging from healthcare to industry 4.0, IoT is growing. It is making the fabric of the environment we live in smarter and responsive, unifying the physical and digital universes. This unification puts forward the need of managing IoT systems to yield greater productivity and efficiency. Workflow automation in IoT helps in efficiently managing IoT systems and enabling effective communication amongst devices.

Nardelli et al. (P2) [33] drew attention toward combining osmotic computing with IoT workflow. They proposed an osmotic flow programming model for mapping as well as executing data generated from IoT transformation workflow applications on a distributed environment integrating cloud and edge. The proposed model was demonstrated through a smart traffic light IoT workflow application. With the leap of changes and benefits provided to the consumers, IoT is extending its branches toward industry. It is making the machines autonomous and connected and benefiting the businesses. However, there are various challenges associated with **Industrial IoT (IIoT)**, one of them is related to data analytics at the network edge (hospital, cars, etc.). Developing analytics solution at the edge brings additional costs and system complexities. To address such challenges a collaborative approach is adopted utilizing available resource-constrained IoT edge analytic hardware. Oyekanlu (P7) [37] presented a software-defined osmotic computing approach to utilize existing low-cost hardware for solving deep learning-based compute-intensive explainable artificial intelligence challenges build on recurrent neural network. It is related to analysis and prediction of future machine data at a IIoT network edge. They proposed an approach to save memory at the edge by constructing coiflet wavelets as microservices for C28x **Digital Signal Processor (DSP)** applying embedded C programming. The author extended this approach to improve cybersecurity in IIoT using a Gaussian Mixture Model. In his work in (P10) [38], a collaborative osmotic approach was presented for many machine learning applications, like novelty and outlier detection in IIoT operation. To show the work ability of the proposed approach, gas turbine machine rotation was considered.

4.3.5 Smart City Applications. Today greater segment of the world is urban population, which demands services and infrastructure such as transport, healthcare, water, and waste management. Information communication technologies play a vital part in developing, deploying, and promoting sustainable development to address growing urbanization needs. Digital technologies meeting the demands of urban cities and the citizens create a smart and sustainable cities. An extensive spectrum of smart city applications ranging from air quality to traffic management is present. Challenges in developing and managing such applications encourages new thinking in various dimensions. Researchers are leveraging the idea of osmotic computing to address the challenges occurring in smart city applications. Filocamo et al. (P9) [18] introduced an osmotic framework for self-adapting traffic in dynamic autonomous vehicular environment. They considered autonomous-vehicle algorithms built on a **vehicle-to-vehicle (V2V)** setup to address the issues occurred during the intersection of automated cars with other vehicles. They proposed a solution using vehicle-to-cloud-edge communication to solve this issue. Different parameters like vehicle speed, position, size, post crossing congestion, and priority of vehicles were considered. The proposed system conformed with the attributes of vehicles and managed priorities in real time. They configured MELs based on the real usecase and deployed them on Raspberry Pi an ESP32.

Souza et al. (P12) [51] adopted osmotic computing principle to create a framework for dynamically balancing workload between edge and cloud. A live migration approach was developed to dynamically deploy microservices across edge and cloud, resulting in efficient utilization of resources. A pilot variant of a microservices-based smart parking application was developed for deployment on an **osmotic execution framework (OEF)**. The architecture of OEF presented a conceptual view to execute MELs in a smart city ecosystem. To validate and evaluate OEF, load tests were carried out. Buzachis et al. (P27) [7] proposed an osmotic flow model leveraging a serverless and light-weight technology: **Functions-as-a-service (FaaS)** to deploy and manage MELs. A video surveillance scenario in smart city was considered to showcase the osmotic model developed. FaaS decomposes applications into triggers and actions (or Functions) and deploys them in provider-managed containers. The osmotic flow model manages, deploys, executs, and migrates MELs across heterogeneous infrastructures, ensuring scalability, flexibility, and reusability. The functionality of the model was validated through several tests and real-world usecases. To improve interoperability, scalabilty, and standardization of smart cities, Pacheco et al. (P13) [39] clubbed deep learning with osmotic computing. They developed a smart lighting prototype for testing the performance of deep learning models on edge/cloud.

4.3.6 MEL Orchestration/Networking. The orchestration and migration of microservices were presented in (P18). Carnevale et al. (P19) [11] investigated the design of an osmotic smart orchestrator for MELs to run IoT applications. Orchestration in osmotic computing-based environment configured the movement/deployment of MELs w.r.t. QoS, security, and privacy needs. A multi-tenant architecture obeying osmotic computing principles was designed to implement a workflow that handles MELs registration, migration, and evaluation. The orchestrator was trained using a deep learning model on the basis of the past experience gathered through the osmotic agent monitoring. The orchestration of MELs in the federated environment is a crucial work that rises various connectivity concerns. Osmotic computing tackles these connectivity issues in the microservices-based federated environment. Buzachis et al. (P20) [8] proposed a network solution to solve connectivity issues in the heterogeneous infrastructure leveraging various network overlays (Calico, Weave, Fannel, OVN). Two different microservices were deployed on FTP and CoAP using container technology. The implemented work was validated through evaluation of network performance with reference to transfer time.

Passing of data (messages) between applications in overlay networks may lead to complexities and security concerns. A **message-oriented middleware (MOM)** approach adds an administrative interface to monitor communication over heterogeneous platforms, optimize responsiveness, and reduce such complexities and security concerns. However, the heterogeneous nature of edge and IoT resources makes the message delivery and QoS optimization complicated. Rausch et al. (P14) [42] borrowed osmotic computing principles and presented an architecture to implement MOM in the heterogeneous infrastructure.

The architecture was build on a pair of diffusion models namely, broker and client diffusion. A server-side messaging protocol (MQTT) was implemented to facilitate message dissemination across the client and broker network. With the increasing complexities over network, the security concerns also increases. Sharma et al. (P22) [46] presented security solutions for media independent handovers and developed an osmotic computing framework using $802.21a - 2012^{TM}$ standard in the dynamic IoT-Fog networks. An osmotic absorption key was used to validate each osmotic layer and point of service. The proposed work solved trust establishment issues and the efficiency was measured in terms of cost and flexibility. An intelligent trust management framework for pervasive online social networks was proposed in (P5) [48]. The approach applied osmotic computing as well as the learning and previous predictions of users' trust for efficient utilization of resources. The framework generated high trust among user and lowered the monitoring cost. The presented work used flexible mixture model and computational offloading concepts to reduce computational time. The lock door policy was considered for trust visualization and identification of trustworthy users. The proposed approach was validated through theoretical and numerical analysis.

4.3.7 Miscellaneous. To improve computational infrastructure and address service heterogeneity in latency-sensitive applications, osmotic computing paradigm plays an insightful role. Sharma et al. (P6) [47] proposed a classification scheme to address service heterogeneity. The services were classified as micro and macro components based on network properties (energy, load, processing time, etc.). A fitness-based algorithm was presented for efficient allocation and swift migration of the classified services across cloud and fog layers. The proposed approach was validated through theoretical and numerical analysis. For efficient handling and management of services, the authors extended their work in (P29) [45]. An osmotic computing framework was proposed for resource scheduling in an augmented reality system. They proposed a service classification approach based on size and bounds. An osmotic game was implemented for efficient scheduling through user and edge initiated osmosis. Kaur et al. (P24) [23] designed an energy-aware framework for efficient resource utilization in an edge/cloud environment. An extended Kalman filter was incorporated for resource monitoring at edge. Migration of microservices was done through Hyper-heuristics. The framework minimized consumption of energy as well as service latency. The proposed approach was validated using container CloudSim with HyFlex on planetlab datasets.

Gamal et al. (P25) [19] presented an osmotic computing approach for load balancing in a distributed environment. The combined osmotic computing principles with bio-inspired algorithms (Ant Colony Optimization and Artificial Bee Colony) for balancing load, VM migration and minimizing energy consumption in cloud. The proposed approach was validated with experimental setups considering fixed and variable loads. Okafor et al. (P4) [36] proposed fog and server consolidation for supporting multi-constrained datacenter environments. A discrete event simulation heuristic was used to evaluate osmotic cloud network for smart grid applications.

4.4 Maturity Level of the Key Topics in Osmotic Computing

Osmotic computing is marching toward advancement since it was introduced in 2016. Many remarkable work has been done as discussed until now. Although, most of the work are in nascent stage. The various key topics of the state of the art: Healthcare, Monitoring, Blockchain, IoT Workflow & IIoT, Smart city Applictions, and MEL orchestration/Networking are adapting emerging technologies and integrating them with osmotic computing principles. However, the maturity level of these key topics is an interesting question to answer, which would direct the researchers toward the possibilities of future directions. The researchers have developed significant usecases and prototypes that are implemented in various HIS, BAN, IoMT, network monitoring, traffic monitoring, smart city monitoring, security leveraging blockchain, cybersecurity, and fault detection using IIoT. However, microservices orchestration and networking are still in the recent stages with their initial works and frameworks yet to be implemented. The research trend analysis of relevant key topics on osmotic computing displayed in Figure 6. This figure indicates that most of the researches are conducted in the recent years.

4.5 Limitation of the Review

A systematic review findings are affected by umpteen factors, like filtering relevant studies from the online databases, biasing in selection of related studies, imprecision in allocating QA score, as well as extraction of data. Finding out all the factors and curating the studies accordingly is an inconvenient task [25]. In this SLR, we explore five openly accessible digital libraries (ACM, IEEE Xplore, ScienceDirect, SpringerLink, and Wiley Online), and along with this we conduct a gray literature search (Google Scholar and ArXiv) to find all the relevant studies. Although,

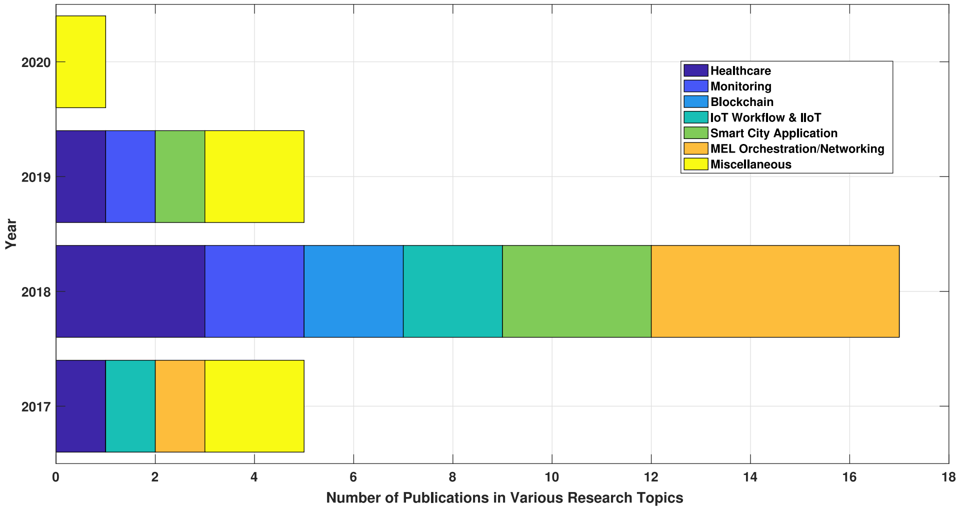

Fig. 6. Research trends of osmotic computing.

some articles could not be included in the review due to their unavailability in any online library. To ensure unbiased selection, we strictly follow our aforementioned systematic review protocols. Based on the research questions, we identify relevant keywords and formulate the query string and feed them to the search engines of different libraries. Each digital library process a different search string, so we define a generic search string.

However, the outcome of the search using generic search string may vary when applied on different digital libraries. A manual advanced search is followed that gives certain extraneous findings that are not in the scope of this SLR. For instance, while conducting search using the search term "osmotic computing," we get several results related to the phenomenon osmosis. Moreover, many articles with insufficient details such as methods and design of the research led to the inaccuracy in allocating quality. We consider articles having highest quality assessment score only. The limitations and discrepancies occurred in this review are settled through discussions between the authors.

5 RESEARCH AND SECURITY CHALLENGES IN OSMOTIC COMPUTING

Villari et al. (P1) [54] have skilfully outlined the research challenges for the osmotic computing paradigm, taking into consideration the future research directions. The research challenges are shown in Table 8. Some of the challenges mentioned in this table are solved, and the work done to solve these challenges are mentioned against them. Along with this, various key issues like resource allocation, scheduling and energy conservation are solved in the mentioned studies. Moreover, there is still scope to improve the work done in various dimensions and considering other major approaches like multi-criteria optimization, multi-criteria decision making to make the the osmotic schedulers more efficient. However, no work has been done to address the microservice resource contention issue. With the advancement in containerization technology, it has become easier to manage resource contention. However, managing resource contention w.r.t. microservices in a federated ecosystem is a crucial task and is yet to be solved. There has been limited but qualitative work done in the domain of osmotic computing that highlights its significance.

Table 8. Research Challenges

Research Challenges	Description	Papers
Microservice Configuration	To develop multi-criteria decision-making frameworks to configure microservices selection across cloud and edge datacenter.	P6, P11, P25, P29
Microservice Networking	To develop consolidated networks for automated management of heterogeneous devices in cloud and edge domain.	P5, P14, P20, P22
Microservice Security	To build a coherent security policy to enable microservice migration across cloud and edge datacenters following same security considerations in both the platforms.	P2, P7, P8, P10, P18
Edge Computing	To categorize microservices on the basis of their characteristics and allocation of appropriate computational resources. To enable microservices migration across clou/edge/IoT.	P5, P6, P16, P17, P21, P26, P28, P29
Microservice Workload Contention	To automatically discover and manage resource contentions among coexisting (codeployed and containerized) microservices across cloud-edge datacenters.	—
Monitoring	To develop monitoring frameworks to monitor workload input along with QoS metrics across microservices on cloud and edge datacenters.	P11, P15, P28
Microservice Orchestration and Elasticity Control	To develop predictive models based on machine learning techniques to estimate workload as well as performance metrics in codeployed microservices across cloud-edge datacenters.	P3, P13, P18, P19, P24, P28

5.1 Security Challenges in Osmotic Computing

With increasing uptake of IoT applications (smart healthcare, smart city, smart home, smart traffic) hosted across (cloud, edge, and IoT) infrastructures, it is observed that services offered by IoT include interrelated providers (such as data, service, network, and infrastructure). Traders in such an environment tacitly anticipate their data as well as services to be reliable and demand to safeguard their privacy. The IoT application users communicate with several applications through a web-based user interface minus any prior knowledge of the federated ecosystem (service, data, and network). The users unknowingly disclose their details to the application providers without thinking that these entrusted providers can give out their details to numerous back-end stakeholders (like Cloud hosted analytics and mobile edge network provider) [40]. The diversification in the computing devices from various vendors and their existence in a distrusted domain without any predominant authorization can raise security challenges for the applications.

Osmotic computing is steered by the remarkable growth in the dimension of resources in a ubiquitous environment, considering deployment, networking, and security of MELs across cloud, edge, and IoT devices. In this dynamic as well as a collaborative system, organizations distribute resources, microservices, and informations to highly scalable applications deployed across cloud/edge/IoT environments. The deployment of microservices in this environment needs efficient security policies to avoid data and privacy protection issues.

Moreover, there is need of a technology that could boost access control to protect digital resources and grant privileges to the users to access them. Blockchain technology (P18) [9] provides a secure and resilient approach for controlling and managing MEL's heterogeneity in an osmotically federated environment through an SDMem [53]. This is the initial work done that proposes orchestration of MEL through an SDMem implementing blockchain facilities with strict adherence to principles of osmotic computing. This primary study is implemented to develop a use case by Villari et al. in (P8) [55].

Security in osmotic computing involves two aspects of MEL orchestration. First, designing and specifying MEL. Second, migrating and managing MEL. However, managing the IoT devices (actuators and servers) in a decentralised system can be taken as a challenge for research.

Moreover, the key concern is security of IoT devices subject to confidentiality, integrity, and authentication of data. IoT has extended toward industrial sectors and applications, referred to as IIoT, which provides industries with efficient and reliable operations by consistently transmitting data to smart devices and machines. IIoT has revolutionized the way industries operate, but boosting digital transformation while maintaining security remains a crucial challenge. Every new device that connects to the IIoT can become a potential liability if the security issues are compromised. Moreover, the cybersecurity measures need to have a stronger hold to eradicate security vulnerabilities.

IIoT requires edge-cloud architecture rather than purely centralized cloud to transform productivity and services in the industrial world. However, in the case of edge computing, many edge nodes can increment footprints of a network that in turn can lead to cybersecurity attacks. Osmotic computing can progressively shift deep learning-based compute-intensive applications (like Apple Siri and Google Now) from cloud to other layers of IoT (edge and fog) (P3) [31]. Oyekanlu in his work (P7) [37] has shown a distributed osmotic perspective for utilizing low-cost hardware to solve complex, compute-intensive deep learning problems across the layers of IIoT systems. His work in (P10) [38] uses osmotic computing principles to develop an IIoT cybersecurity and fault-monitoring applications to find inconsistencies in IIoT machines. However, with applications achieving greater heights, security becomes monotonous target to reach. Moreover, extensive security directions for advance research in osmotic computing are yet to be explored.

6 CONCLUSION

In this article, we present an SLR on osmotic computing. This review follows a systematic process and adheres to the guidelines discussed in repute publications. We extract relevant studies from various digital libraries and gray literature based on a generic search query. Thereafter, we screen the articles by following various selection criteria and remove irrelevant articles. The quality of each relevant study is assessed against the QA questions. We select 29 fundamental articles from 99 articles available in the digital libraries published during the years 2016–2020. We present an overview of osmotic computing along with its relation with various paradigms like Cloud, fog, and edge computing. Next, we provide a summary of relevant work done on osmotic computing and outline the features and research challenges related to it. Finally, we discuss the security issues resolved and unresolved in osmotic computing.

In this review, we conclude that osmotic computing paradigm is a growing research domain and many challenges are not yet adequately tackled, so there are many possibilities for future research on osmotic computing issues. The maturity level of the studies done in this field is low, and most of the works are initial work and usecases/prototypes.

Thus, promising work can be done to convert the usecases/prototypes into real-time usecases. Moreover, most of the work done in this field use machine learning algorithms, but very few such algorithms are explored. Hence, there is scope to use different other approaches and explore other machine learning algorithms in osmotic computing applications.

APPENDIX

A QUALITY ASSESSMENT RESULTS

The standard of the papers considered in this SLR is assessed properly through several QA questions mentioned in Section 2. The quality assessment results are presented in Table 9.

Table 9. Quality Assessment Results

PID	Quality Assessment Score					Total	PID	Quality Assessment Score					Total
	QA_1	QA_2	QA_3	QA_4	QA_5			QA_1	QA_2	QA_3	QA_4	QA_5	
P1	1	1	0	1	1	4	P16	1	1	1	1	1	5
P2	1	0.5	0	1	1	3.5	P17	1	0.5	1	1	1	4.5
P3	0.5	0.5	0	1	1	3	P18	1	0.5	1	1	1	4.5
P4	1	0.5	1	1	1	4.5	P19	1	0	1	1	0.5	3.5
P5	1	0.5	1	1	1	4.5	P20	1	0.5	1	1	1	4.5
P6	1	1	0	1	1	4	P21	1	0.5	1	1	0.5	4
P7	1	0	0.5	1	1	3.5	P22	1	0.5	0	1	1	3.5
P8	0.5	0	1	1	0.5	3	P23	1	0.5	0	1	0.5	3
P9	0.5	0	1	1	0.5	3	P24	1	0.5	0	1	1	3.5
P10	1	0.5	0.5	1	1	4	P25	1	1	1	1	1	5
P11	1	0.5	1	1	1	4.5	P26	1	1	1	1	1	5
P12	1	1	1	1	1	5	P27	1	0.5	1	1	1	4.5
P13	1	0.5	0	1	1	3.5	P28	1	1	1	1	1	5
P14	1	0.5	0	1	0.5	3	P29	1	1	0	1	1	4
P15	1	0.5	1	1	1	4.5							

Satisfactory = 1; Partially Satisfactory = 0.5; Unsatisfactory = 0.

REFERENCES

[1] Muhammad Ali Babar and He Zhang. 2009. Systematic literature reviews in software engineering: Preliminary results from interviews with researchers. In *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM'09)*. IEEE Computer Society, 346–355. https://doi.org/10.1109/ESEM.2009.5314235

[2] D. Bernstein. 2014. Containers and cloud: From LXC to docker to kubernetes. *IEEE Cloud Comput.* 1, 3 (Sep. 2014), 81–84. https://doi.org/10.1109/MCC.2014.51

[3] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the Internet of Things. In *Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing (MCC'12)*. 13–16.

[4] Pearl Brereton, Barbara A. Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. 2007. Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw.* 80, 4 (2007), 571–583. https://doi.org/10.1016/j.jss.2006.07.009

[5] A. Buzachis, G. M. Bernava, M. Busa, G. Pioggia, and M. Villari. 2018. Towards osmotic computing: Future prospect for the health information technology (HIT) systems of ISASI-CNR (ME). In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC'18)*. 01255–01260. https://doi.org/10.1109/ISCC.2018.8538714

[6] A. Buzachis, G. M. Bernava, M. Busa, G. Pioggia, and M. Villari. 2018. Towards the basic principles of osmotic computing: A closed-loop gamified cognitive rehabilitation flow model. In *Proceedings of the IEEE 4th International Conference on Collaboration and Internet Computing (CIC'18)*. 446–452. https://doi.org/10.1109/CIC.2018.00067

[7] Alina Buzachis, Maria Fazio, Antonio Celesti, and Massimo Villari. 2019. Osmotic flow deployment leveraging FaaS capabilities. In *Internet and Distributed Computing Systems*, Raffaele Montella, Angelo Ciaramella, Giancarlo Fortino, Antonio Guerrieri, and Antonio Liotta (Eds.). Springer International Publishing, Cham, 391–401.

[8] A. Buzachis, A. Galletta, L. Carnevale, A. Celesti, M. Fazio, and M. Villari. 2018. Towards osmotic computing: Analyzing overlay network solutions to optimize the deployment of container-based microservices in fog, edge and IoT environments. In *Proceedings of the IEEE 2nd International Conference on Fog and Edge Computing (ICFEC'18)*. 1–10. https://doi.org/10.1109/CFEC.2018.8358729

[9] A. Buzachis and M. Villari. 2018. Basic principles of osmotic computing: Secure and dependable microelements (MELs) orchestration leveraging blockchain facilities. In *Proceedings of the IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion'18)*. 47–52. https://doi.org/10.1109/UCC-Companion.2018.00033

[10] Charles C. Byers and Patrick Wetterwald. 2015. Fog computing distributing data and intelligence for resiliency and scale necessary for IoT: The Internet of Things (ubiquity symposium). *Ubiquity* 2015, Article 4 (Nov. 2015), 12 pages. https://doi.org/10.1145/2822875

[11] L. Carnevale, A. Celesti, A. Galletta, S. Dustdar, and M. Villari. 2018. From the cloud to edge and IoT: A smart orchestration architecture for enabling osmotic computing. In *Proceedings of the 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA'18)*. 419–424. https://doi.org/10.1109/WAINA.2018.00122

[12] Lorenzo Carnevale, Antonio Celesti, Antonino Galletta, Schahram Dustdar, and Massimo Villari. 2019. Osmotic computing as a distributed multi-agent system: The body area network scenario. *Internet Things* 5 (2019), 130–139. https://doi.org/10.1016/j.iot.2019.01.001

[13] Lorenzo Carnevale, Antonino Galletta, Antonio Celesti, Maria Fazio, Maurizio Paone, Placido Bramanti, and Massimo Villari. 2018. Big data HIS of the IRCCS-ME future: The osmotic computing infrastructure. In *Cloud Infrastructures, Services, and IoT Systems for Smart Cities*, Antonella Longo, Marco Zappatore, Massimo Villari, Omer Rana, Dario Bruneo, Rajiv Ranjan, Maria Fazio, and Philippe Massonet (Eds.). Springer International, Cham, 199–207.

[14] Christopher Carroll and Andrew Booth. 2015. Quality assessment of qualitative evidence for systematic review and synthesis: Is it meaningful, and if so, how should it be performed? *Res. Synth. Methods* 6, 2 (2015), 149–154. https://doi.org/10.1002/jrsm.1128

[15] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. 2011. Clonecloud: Elastic execution between mobile device and cloud. In *Proceedings of the 6th Conference on Computer Systems*. 301–314.

[16] David Smith Ed Anderson, Leah Ciavardini. 2019. *Hype Cycle for Cloud Computing*.

[17] M. Fazio, A. Celesti, R. Ranjan, C. Liu, L. Chen, and M. Villari. 2016. Open issues in scheduling microservices in the cloud. *IEEE Cloud Comput.* 3, 5 (Sep. 2016), 81–88. https://doi.org/10.1109/MCC.2016.112

[18] B. Filocamo, A. Galletta, M. Fazio, J. A. Ruiz, M. A. Sotelo, and M. Villari. 2018. An innovative osmotic computing framework for self adapting city traffic in autonomous vehicle environment. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC'18)*. 01267–01270. https://doi.org/10.1109/ISCC.2018.8538675

[19] M. Gamal, R. Rizk, H. Mahdi, and B. E. Elnaghi. 2019. Osmotic bio-inspired load balancing algorithm in cloud computing. *IEEE Access* 7 (2019), 42735–42744. https://doi.org/10.1109/ACCESS.2019.2907615

[20] Julian P. T. Higgins, James Thomas, Jacqueline Chandler, Miranda Cumpston, Tianjing Li, Matthew J. Page, and Vivian A. Welch. 2019. *Cochrane Handbook for Systematic Reviews of Interventions* (2nd ed.). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119536604

[21] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. 2015. Mobile edge computing—A key technology towards 5G. *ETSI White Paper* 11, 11 (2015), 1–16.

[22] P. Jamshidi, C. Pahl, N. C. Mendonca, J. Lewis, and S. Tilkov. 2018. Microservices: The journey so far and challenges ahead. *IEEE Softw.* 35, 3 (May 2018), 24–35. https://doi.org/10.1109/MS.2018.2141039

[23] Kuljeet Kaur, Sahil Garg, Georges Kaddoum, Syed Hassan Ahmed, and Dushantha Nalin K. Jayakody. 2019. En-OsCo: Energy-aware osmotic computing framework using hyper-heuristics. In *Proceedings of the ACM MobiHoc Workshop on Pervasive Systems in the IoT Era (PERSIST-IoT'19)*. ACM, New York, NY, 19–24. https://doi.org/10.1145/3331052.3332473

[24] Barbara Kitchenham and Pearl Brereton. 2013. A systematic review of systematic review process research in software engineering. *Inf. Softw. Technol.* 55, 12 (Dec. 2013), 2049–2075. https://doi.org/10.1016/j.infsof.2013.07.010

[25] Barbara Kitchenham, Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering—A systematic literature review. *Inf. Softw. Technol.* 51, 1 (2009), 7–15. https://doi.org/10.1016/j.infsof.2008.09.009

[26] Barbara Kitchenham, Rialette Pretorius, David Budgen, O. Pearl Brereton, Mark Turner, Mahmood Niazi, and Stephen Linkman. 2010. Systematic literature reviews in software engineering—A tertiary study. *Inf. Softw. Technol.* 52, 8 (Aug. 2010), 792–805. https://doi.org/10.1016/j.infsof.2010.03.006

[27] A. Longo, A. De Matteis, and M. Zappatore. 2018. Urban pollution monitoring based on mobile crowd sensing: An osmotic computing approach. In *Proceedings of the IEEE 4th International Conference on Collaboration and Internet Computing (CIC'18)*. 380–387.

[28] Antonella Longo, Marco Zappatore, and Andrea De Matteis. 2019. An osmotic computing infrastructure for urban pollution monitoring. *Softw.: Pract. Exper.* (2019), 1–25. https://doi.org/10.1002/spe.2721 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.2721

[29] Redowan Mahmud, Ramamohanarao Kotagiri, and Rajkumar Buyya. 2018. Fog computing: A taxonomy, survey and future directions. In *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*, Beniamino Di Martino, Kuan-Ching Li, Laurence T. Yang, and Antonio Esposito (Eds.). Springer Singapore, 103–130. https://doi.org/10.1007/978-981-10-5861-5_5

[30] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G. Altman, and The PRISMA Group. 2009. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLOS Med.* 6, 7 (Jul. 2009), 1–6. https://doi.org/10.1371/journal.pmed.1000097

[31] A. Morshed, P. P. Jayaraman, T. Sellis, D. Georgakopoulos, M. Villari, and R. Ranjan. 2017. Deep osmosis: Holistic distributed deep learning in osmotic computing. *IEEE Cloud Comput.* 4, 6 (Nov. 2017), 22–32. https://doi.org/10.1109/MCC.2018.1081070

[32] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan. 2018. Fog computing: Survey of trends, architectures, requirements, and research directions. *IEEE Access* 6 (2018), 47980–48009. https://doi.org/10.1109/ACCESS.2018.2866491

[33] M. Nardelli, S. Nastic, S. Dustdar, M. Villari, and R. Ranjan. 2017. Osmotic flow: Osmotic computing + IoT workflow. *IEEE Cloud Comput.* 4, 2 (Mar. 2017), 68–75. https://doi.org/10.1109/MCC.2017.22

[34] Rim Negra, Imen Jemili, and Abdelfettah Belghith. 2016. Wireless body area networks: Applications and technologies. *Proc. Comput. Sci.* 83 (2016), 1274–1281. https://doi.org/10.1016/j.procs.2016.04.266

[35] Mark Newman and David Gough. 2020. Systematic reviews in educational research: Methodology, perspectives and application. In *Systematic Reviews in Educational Research: Methodology, Perspectives and Application*, Olaf Zawacki-Richter, Michael Kerres, Svenja Bedenlier, Melissa Bond, and Katja Buntins (Eds.). Springer Fachmedien Wiesbaden, 3–22. https://doi.org/10.1007/978-3-658-27602-7_1

[36] K. C. Okafor, F. N. Ugwoke, and A. A. Obayi. 2017. Evaluation of virtualized osmotic cloud network using discrete event Branch-and-Bound heuristics. In *Proceedings of the IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON'17)*. 425–437. https://doi.org/10.1109/NIGERCON.2017.8281912

[37] E. Oyekanlu. 2018. Distributed osmotic computing approach to implementation of explainable predictive deep learning at industrial IoT network edges with real-time adaptive wavelet graphs. In *Proceedings of the IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE'18)*. 179–188. https://doi.org/10.1109/AIKE.2018.00042

[38] E. Oyekanlu. 2018. Osmotic collaborative computing for machine learning and cybersecurity applications in industrial IoT networks and cyber physical systems with gaussian mixture models. In *Proceedings of the IEEE 4th International Conference on Collaboration and Internet Computing (CIC'18)*. 326–335. https://doi.org/10.1109/CIC.2018.00051

[39] A. Pacheco, P. Cano, E. Flores, E. Trujillo, and P. Marquez. 2018. A smart classroom based on deep learning and osmotic IoT computing. In *Proceedings of the Congreso Internacional de Innovacion y Tendencias en Ingenieria (CONIITI'18)*. 1–5. https://doi.org/10.1109/CONIITI.2018.8587095

[40] D. Puthal, S. Nepal, R. Ranjan, and J. Chen. 2016. Threats to networking cloud and edge datacenters in the Internet of Things. *IEEE Cloud Comput.* 3, 3 (May 2016), 64–71. https://doi.org/10.1109/MCC.2016.63

[41] R. Ranjan, O. Rana, S. Nepal, M. Yousif, P. James, Z. Wen, S. Barr, P. Watson, P. Jayaraman, D. Georgakopoulos, M. Villari, M. Fazio, S. Garg, R. Buyya, L. Wang, A. Y. Zomaya, and S. Dustdar. 2018. The next grand challenges: Integrating the Internet of Things and data science. *IEEE Cloud Comput.* 5, 03 (Jan. 2018), 12–26. https://doi.org/10.1109/MCC.2018.032591612

[42] T. Rausch, S. Dustdar, and R. Ranjan. 2018. Osmotic message-oriented middleware for the Internet of Things. *IEEE Cloud Comput.* 5, 02 (Jan. 2018), 17–25. https://doi.org/10.1109/MCC.2018.022171663

[43] M. Satyanarayanan. 2017. The emergence of edge computing. *Computer* 50, 1 (Jan. 2017), 30–39. https://doi.org/10.1109/MC.2017.9

[44] Joachim Schopfel and Dominic J. Farace. 2010. Grey literature. In *Encyclopedia of Library and Information Sciences* (3rd ed.). CRC Press, 2029–2039.

[45] Vishal Sharma, Dushantha Nalin K. Jayakody, and Marwa Qaraqe. 2020. Osmotic computing-based service migration and resource scheduling in mobile augmented reality networks (MARN). *Fut. Gener. Comput. Syst.* 102 (2020), 723–737. https://doi.org/10.1016/j.future.2019.09.008

[46] Vishal Sharma, Jiyoon Kim, Soonhyun Kwon, Ilsun You, and Fang-Yie Leu. 2018. An overview of 802.21a-2012 and its incorporation into IoT-fog networks using osmotic framework. In *IoT as a Service*, Yi-Bing Lin, Der-Jiunn Deng, Ilsun You, and Chun-Cheng Lin (Eds.). Springer International Publishing, Cham, 64–72.

[47] Vishal Sharma, Kathiravan Srinivasan, Dushantha Nalin K. Jayakody, Omer F. Rana, and Ravinder Kumar. 2017. Managing service-heterogeneity using osmotic computing. arxiv:1704.04213. Retrieved from http://arxiv.org/abs/1704.04213.

[48] Vishal Sharma, Ilsun You, Ravinder Kumar, and Pankoo Kim. 2017. Computational offloading for efficient trust management in pervasive online social networks using osmotic computing. *IEEE Access* 5 (2017), 5084–5103.

[49] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. 2016. Edge computing: Vision and challenges. *IEEE IoT J.* 3, 5 (Oct. 2016), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

[50] A. Souza, N. Cacho, A. Noor, P. P. Jayaraman, A. Romanovsky, and R. Ranjan. 2018. Osmotic monitoring of microservices between the edge and cloud. In *Proceedings of the IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS'18)*. 758–765. https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00129

[51] A. Souza, Z. Wen, N. Cacho, A. Romanovsky, P. James, and R. Ranjan. 2018. Using osmotic services composition for dynamic load balancing of smart city applications. In *Proceedings of the IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA'18)*. 145–152. https://doi.org/10.1109/SOCA.2018.00029

[52] Massimo Villari, Antonio Celesti, and Maria Fazio. 2018. Towards osmotic computing: Looking at basic principles and technologies. In *Complex, Intelligent, and Software Intensive Systems*, Leonard Barolli and Olivier Terzo (Eds.). Springer International Publishing, Cham, 906–915.

[53] M. Villari, M. Fazio, S. Dustdar, O. Rana, L. Chen, and R. Ranjan. 2017. Software defined membrane: Policy-driven edge and Internet of Things security. *IEEE Cloud Comput.* 4, 04 (Jan. 2017), 92–99. https://doi.org/10.1109/MCC.2017.3791014

[54] M. Villari, M. Fazio, S. Dustdar, O. Rana, and R. Ranjan. 2016. Osmotic computing: A new paradigm for edge/cloud integration. *IEEE Cloud Comput.* 3, 6 (Nov. 2016), 76–83. https://doi.org/10.1109/MCC.2016.124

[55] M. Villari, A. Galletta, A. Celesti, L. Carnevale, and M. Fazio. 2018. Osmotic computing: Software defined membranes meet private/federated blockchains. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC'18)*. 01292–01297. https://doi.org/10.1109/ISCC.2018.8538546

Received July 2020; revised September 2021; accepted September 2021