Mobility-aware fog computing in dynamic networks with mobile nodes: A survey

Krzysztof Ostrowski a,∗, Krzysztof Małecki a, Piotr Dziurzański a, Amit Kumar Singh b

a West Pomeranian University of Technology, Szczecin, Poland
b University of Essex, Colchester, UK

A R T I C L E   I N F O

Keywords:
Mobile fog computing
Mobility-aware
Dynamic networks

A B S T R A C T

Fog computing is an evolving paradigm that addresses the latency-oriented performance and spatio-temporal issues of the cloud services by providing an extension to the cloud computing and storage services in the vicinity of the service requester. In dynamic networks, where both the mobile fog nodes and the end users exhibit time-varying characteristics, including dynamic network topology changes, there is a need of mobility-aware fog computing, which is very challenging due to various dynamisms, and yet systematically uncovered. This paper presents a comprehensive survey on the fog computing compliant with the OpenFog (IEEE 1934) standardised concept, where the mobility of fog nodes constitutes an integral part. A review of the state-of-the-art research in fog computing implemented with mobile nodes is conducted. The review includes the identification of several models of fog computing concept established on the principles of opportunistic networking, social communities, temporal networks, and vehicular ad-hoc networks. Relevant to these models, the contributing research studies are critically examined to provide an insight into the open issues and future research directions in mobile fog computing research.

1. Introduction

With the proliferation of heterogeneous devices (the things) connected to the Internet of Things (IoT), i.e. networks of sensors and actuators (ITU-T, 2012), the overall volume and velocity of data generated and consumed in IoT networks increases rapidly. Conventional cloud-based centralised solutions deployed in far-end data centres of theoretically unlimited location-independent computing and storage resources, Armbrust et al. (2010) and Mell and Grance (2011), impose significant disadvantages if applied directly to the IoT environment (Yannuzzi et al., 2014; Al-Fuqaha et al., Fourthquarter 2015; Chiang and Zhang, 2016). A significant rise in latency and energy consumption introduced by transfers of data produced by geographically distributed IoT end-users to a centralised cloud is one of the very first observable issues. Furthermore, conventional cloud-based centralised solutions suffer from irrecoverable losses in contextual information that may result in inappropriate judgements based on the situations deduced from the well-formed but no longer valid data. Moreover, a cloud-oriented management of a multitude of constrained end devices of intermittent network connectivity that affects the IoT network’s scalability, privacy, and security poses serious challenges. To address the listed issues, several computing paradigms were developed over the years to offer the cloud services, storage and processing resources “closer” to the service requester as represented by an IoT device (Yousefpour et al., 2019). The vast majority of the researched solutions fall into one of two categories of the conceptual computing paradigms, edge computing, and fog computing.

Edge computing is understood as a paradigm that introduces cloud services at the edge of the network by means of network nodes in the proximity of the service requester, Satyanarayanan (2017). Furthermore, edge computing predates the formulation of fog computing, or even cloud computing (as found in a form of distributed data caching services in Dilley et al. (2002), Xie et al. (2002) and Davis et al. (2004)). Edge computing includes a spectrum of concepts, e.g. cloudlet which focuses on microdata centres (Satyanarayanan et al., 2009), or mobile cloud computing (MCC) which delegates tasks from mobile phones to cloudlets (Kumar and Lu, 2010; Dinh et al., 2013; Bou Abdo and Demerjian, 2017), as well as some implementations, which include multi-access edge computing (MEC), formerly known as mobile edge computing. MEC leverages the existing cellular networks infrastructure, and it is deemed to be a step in the evolution of mobile base stations (Hu et al., 2015; ETSI, 2019).
Fog computing (FC) is a conceptual computing paradigm and a standardised system-level hierarchical concept architecture (IEEE, 1934) for a highly virtualised platform that distributes computation, storage, and networking tasks over the autonomous and heterogeneous fog nodes to assist and extend the cloud infrastructure for IoT devices (Bonomi et al., 2012; Byers, 2017). Fig. 1 depicts the context for the standardised fog computing, where the heterogeneity and both the qualitative and quantities granularities of the system actors are emphasised. In fact, any fog-capable device can act as a network node in a fog computing environment (i.e. as a fog node) and actively provide services to end-users and other fog nodes (Bonomi et al., 2014). A fog system may be either a network of connected systems arranged into hierarchies of nodes with controlled lifetimes or a solution folded into a single physical system. Multi-tier computing is at the heart of fog computing. A significant variability in the location, the resulting networking latencies, and the characteristics of fog nodes are visualised in Fig. 1(a), which illustrates the fog computing system with the example network devices that take part in building the cloud-to-thing continuum along and their quantities. As opposed to edge computing, in fog computing, the network nodes can be located anywhere between...
the cloud and the edge of the network, and benefit from a wide diversity of connected devices (Chiang et al., 2017). Heterogeneity at multiple levels (e.g. device capabilities, network connection, etc.) is inherent in the fog computing paradigm. It may be seen as its primary strength in building distributed systems that integrate things and the cloud. The multi-layered fog network design promotes cooperation with a cloud which, in turn, constitutes a cloud-to-thing continuum, where fog computing is thought of as an extension of cloud-based models that preserves all the benefits brought by cloud computing. Fig. 1(b) figuratively presents the cloud-to-thing continuum composed out of a diversity of interconnected nodes. In particular, fog relies on the cloud and edge ideas to bridge the gap between things and the cloud. Thus it may be seen as the highest evolution of the edge computing principles (De Donno et al., 2019).

Notably, fog nodes are given a high degree of autonomy, so that they are free to actively collaborate with each other and the things, perform analytics to take local decisions independently and efficiently, manage applications, and react to security threats without the involvement of the network operator, or even spontaneously change their location due to node mobility (IEEE, 1934). Recent works in fog computing take advantage of autonomic computing (Kephart and Chess, 2003; Ganek and Corbi, 2003), to cope with the autonomy aspect of fog environments (Tabir et al., 2019; Menascé, 2020). Autonomic computing is founded on the idea of a control loop that monitors the environment. It acts according to adaptable policies that limit the set of decisions and actions the autonomic entity performs (this is known as the monitor-analyse-plan-execute-knowledge, MAPE-K, framework) (IBM, 2006). Coalescing of independent judgements seems to be a major concern in building robust fog solutions.

A reference architecture for fog computing was initially provided by OpenFog Consortium, and later adopted as IEEE (1934) in 2018. The IEEE adoption of the OpenFog reference architecture states that edge computing was originally defined with the exclusion of a cloud. In contrast, fog computing was devised as an extension of a cloud. That implies those two concepts are distinct, and, at most, fog is a superset of edge computing (Junior et al., 2021). While the general objectives like, i.a. real-time responsiveness, location awareness, and mobility support are shared between fog and edge, notable differences exist that have a foundation in approach to diversity and location of nodes, as well as in solving network control plane management.

A fog computing network that solely includes infrastructure nodes (i.e. those of fixed locations) manifests mobility in the spatio-temporal behaviour of a service requester (cf. MEC). A ubiquitous computing solution brought by the fog leverages the mobility of nodes in the fog environment, where the mobility concern also applies to adventitious location changes of the service requester node itself. Two directions in mobility-aware fog computing research are leading recently: mobile fog computing (Chang et al., 2020; Gima et al., 2020), and vehicular fog computing (Ostrowski and Malecki, 2023; Ning et al., 2019). Researchers ponder three mobility perspectives: mobility of a service requester, fog nodes, or both. Fog computing is expected to address all of them.

1.1. Dynamic networks

Networks in which nodes may leave and join at any time and where the network connections may disappear and recover as time passes are termed as dynamic networks (Rossetti et al., 2017). Current research studies provide tools to describe and analyse such networks, i.e. temporal networks theory and opportunistic computing paradigm (Newman, 2003; Rossetti and Cazabet, 2018; Conti et al., 2010).

Temporal networks theory, an extension of complex networks, is a mathematical framework to model dynamic structures with relations and interactions (Holme and Saramäki, 2012). Modelling of fog networks as temporal ones provides a tool to identify and track the mobile fog nodes in relationships (Cazabet and Rossetti, 2019), which is essential to ensure the service continuity property in highly dynamic environments. To the best of our knowledge, none of the related existing surveys neither engaged nor explored the temporal network perspective of fog computing.

The opportunistic computing paradigm evolved from the idea of opportunistic networks, the networks that enable node communication even in the absence of a route between the nodes, to an environment that provides general computing services (Pelusi et al., 2006; Conti et al., 2015). Opportunistic networking exploits the self-organising networks of unstable topologies and high churn rates with locally executed contextual nodal decisions (e.g. routing, scheduling, etc.). Computing services laid over those networks are seen as roots of fog computing (Das, 2018; Mascitti et al., 2018) that employ complex network theory to reason about (Conti and Passarella, 2018). In contrast to this survey, the related existing surveys predominantly describe the mobile user perspective while discussing mobility in fog computing. Moreover, none of those works strictly discusses the mobility of computing fog nodes, neither discover various models of fog computing that share the opportunistic foundation. Note that UAV-assisted solutions constraint the mobility with pre-planned moving dynamics and strictly defined flying paths (Motalagh et al., 2019; Zhu et al., 2021b) which effectively preclude self-decided node mobility and, in turn, factor out UAV research from this survey.

1.2. Contribution

The existing surveys emphasise the mobility of a service requester as a main determinant of mobility-awareness, while this publication tackles the mobility problem from the perspective of fog network dynamics, where fluctuations in the network topology are natural. As found out during careful examination, the available review articles tend to point out differences rather than search for commonalities in the identified models of fog computing, its predecessors (e.g. where fog computing is the most general and complete representative Yousefpour et al., 2019), and in the discovered similar ideas. Few of the latest studies use the pillars and the concept defined by OpenFog as evaluation criteria (Bellavista et al., 2019; Puliafito et al., 2019), and most works are unaware of them. Nevertheless, multiple of them only refer to the arbitrarily selected properties. While most of the papers describe solutions to a problem of the end-user mobility (i.e. when IoT device changes its geographical or topological location), or do not state the used mobility perspective at all, only three of them actually clearly distinguish between network and user mobility perspectives (Mouradian et al., 2018; Mahmud et al., 2020; Gill and Singh, 2021). Interestingly, nearly all the examined surveys emphasise the cruciality of further research in the mobility-awareness aspect of fog computing. To the best of our knowledge, none of the available surveys in fog computing directly addresses the node mobility concerns in mobile fog computing environments. Furthermore, this paper puts emphasis on the relationship between node mobility and the application (i.e. sets of executed tasks) level, thus, elaborations on models of communication channels and related signal processing aspects are minimal.

According to the authors’ knowledge, there exists no survey publication entirely dedicated to mobility-awareness in fog computing that provides a complete view on that aspect under the umbrella of the fog computing concept and its models. Furthermore, the available survey publications briefly mention the related mobility issues while discussing other characteristics of fog computing. Therefore, the authors argue that this survey is the very first attempt to systematically review the research efforts in fog computing with mobile fog nodes. A detailed evaluation of the existing surveys is provided in Section 2.

In this survey, a critical evaluation of mobility-aware solutions that contribute to the models of fog computing architecture with mobile computing nodes is presented. The main contributions are as follows:
(1) Rely on well-established definitions. Highlight mobility concerns that are expressed in the standardised architecture complaint with IEEE 1934.

(2) Apply a synergistic approach. Identify models of the fog computing concept that contribute to the mobility pillars, and which combination leads to a robust mobility-aware fog solution.

(3) Find the mature building blocks. Provide a comprehensive insight into the latest research efforts related to mobility-aware models of fog computing focused on computing network mobility in particular.

(4) Provide directions. Present open issues and research opportunities.

The foundations of the review criteria rely on the several key ingredients of opportunistic networking and computing applicable to OpenFog definition of fog computing (e.g. nodal collaboration, mobility management, resource management, etc.) considered in the environment of dynamic networks. Examined research works include publications from years 2021, 2020, 2019, and earlier with percentage quantitative contributions of 39%, 34%, 20%, and 7%, respectively.

The remainder of this article is organised as follows. Section 3 provides details of the fog computing concept focused on the mobility-related features. In Section 4, models of the mobility-aware fog computing concept architecture are described. Related surveys are presented in Section 2. Section 5 categorises existing research works in the area of mobile fog computing by the fog computing models and the addressed problems. Identified open issues and research challenges are discussed in Section 6. Finally, Section 7 concludes this article. Fig. 2 illustrates the relationships between the sections.

2. Related surveys

This section summarises current review articles that consider the mobility facet of fog computing. The selection of the survey publications covers the related works available in June, 2022. Detailed screening of the papers led to the final quality-oriented selection of the survey papers from the initially qualified ones. Table 1 summarises the evaluated existing 15 surveys in fog computing that allude to aspect of mobility. A work is OpenFog-complaint if it directly references the standardised architecture, applies the in there settled terminology, and conducts considerations according to the framework defined therein.

Authors of Mahmud et al. (2018) identify mobile computing nodes in vehicular setups only. Discussions hinged on OpenFog reference architecture for fog computing are found in Bellavista et al. (2019) and Puliafito et al. (2019). Authors of Bellavista et al. (2019) point out the impact of mobility on the communication, and briefly present mobility-aware protocols for resource discovery and routing. Mobility-sensitive system components (e.g. migration) are discussed from the angle of the end-user mobility. Similarly, Puliafito et al. (2019) discuss the impact of the end user mobility on the fog platforms and migration policies. However, the moving computing nodes are not taken into account.

A work in Rejiba et al. (2019) outlines several possible migration scenarios that consider both the end user and node mobility, as well as the migration assisted by infrastructure nodes. The authors stress the role of nodal cooperation as an important factor in providing complete migration solutions. Migration in conjunction with mobility of nodes, and consequently, of nodes and users, is not covered.

A comparison of selected constrained models of the fog computing concept is provided as either a background study or as a basis to point out the future directions of fog computing evolution in Yousefpour et al. (2019). The exploitation of the opportunistic computing to further decentralise and manage the dynamic networks is not covered. The mobility perspective is not explicitly specified.

Authors of Mahmud et al. (2020) elaborate on a logically distributed application management. Mobility aspects are discussed in the event-driven iterative application placement strategy that is sensitive to node failures and mobility. Solutions that reflect the dynamic changes of an application architecture in the fog network topology are not discussed.

Work in Salaht et al. (2020) presents a literature review in solutions to the online service placement with distributed orchestration. Mobility is defined as a characteristic of a placement strategy which, in turn, is materialised in the form of a service migration, and discussed from...
Due to standardised formulation of the fog computing as a concept, i.e. an abstract notion, there exist various models of fog computing that strive to strip away layers of its complexity in order to materialise bases of computing, and similar concepts.

The categorisation of the sources of context-awareness given in Islam et al. (2021) classifies mobility as an inherent characteristic of fog computing that originates from multiple levels of the contextual knowledge. Contexts of a user, device, and environment are marked as crucial in the mobility management. The authors do not reference the standardised fog architecture and discuss the end user mobility only.

Authors of Costa et al. (2023) notice the dynamicity of the fog computing environments and enumerate its impact on fog orchestration. However, they do not discuss setups with predominant mobile fog nodes. Recent works loosely refer to OpenFog while discussing fog computing in smart cities (Songhorabadi et al., 2023), however, they do not clearly emphasise the value of mobile computing nodes (Songhorabadi et al., 2023; Bukhari et al., 2022).

3. Mobility in the fog computing concept

IEEE 1934’s fog computing is built on nine pillars: security, manageability, scalability, openness, autonomy, reliability-availability-serviceability (RAS), agility, hierarchy, programmability (IEEE, 1934). While all the pillars are crucial in the delivery of a complete fog computing network, few of them play pivotal role in the mobility-awareness characteristics, namely: autonomy, agility, manageability, scalability, and RAS (Fig. 3a).

Autonomy is an essential enabler for decision-making, in the decentralised and distributed forms, across the layers of the hierarchical network. Autonomic behaviour enables fog nodes to cooperate, create the intelligence at the network edge, monitor the environment, and react promptly to ensure uninterrupted operations. Agility provides a context (Dey, 2001) out of the data generated by things and fog nodes, effectively driving context-aware decisions that deal with dynamic nature of fog deployments. Agile fog networking aims to respond quickly to handovers and migration of services (Pollini, 1996; Stemm and Katz, 1998). User mobility is well studied by researchers. In contrast, mobility in the fog computing network introduces extensive changes to the computing environment on multiple levels which, in turn, trigger various operations starting from the recovery of the network links (physical connection channels), through multi-hop routing to rebuild the node communication graph, along with re-provisioning of the workload according to the offloaded application characteristics, and ending with the migration of services to candidate nodes proposed by the service discovery feature. Spontaneous node mobility poses serious challenges, but also offers great opportunities to fog systems.

4. Mobility-aware models of fog computing

<table>
<thead>
<tr>
<th>Reference</th>
<th>Year</th>
<th>Area</th>
<th>OpenFog compliant</th>
<th>Mobility perspective</th>
<th>Identified models of fog computing, and similar concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mahmud et al. (2018)</td>
<td>2018</td>
<td>comp. paradigms and taxonomy</td>
<td>No</td>
<td>User</td>
<td>MEC, MCC</td>
</tr>
<tr>
<td>Mouradian et al. (2018)</td>
<td>2018</td>
<td>System arch., algorithms</td>
<td>Partially</td>
<td>User, network</td>
<td>Cloudlet, MEC</td>
</tr>
<tr>
<td>Bellavista et al. (2019)</td>
<td>2019</td>
<td>IoT-oriented fog solutions</td>
<td>Yes</td>
<td>User</td>
<td>--</td>
</tr>
<tr>
<td>Puliafito et al. (2019)</td>
<td>2019</td>
<td>Software and hardware platforms</td>
<td>Yes</td>
<td>User</td>
<td>MEC, cloudlet, MEC</td>
</tr>
<tr>
<td>Rejiba et al. (2019)</td>
<td>2019</td>
<td>Migration</td>
<td>Yes</td>
<td>User, network</td>
<td>Cloudlet, MEC, vehicular clouds</td>
</tr>
<tr>
<td>Yonsefpoor et al. (2019)</td>
<td>2019</td>
<td>Multiple (general overview)</td>
<td>Partially</td>
<td>Mixed</td>
<td>MEC, cloudlet, (others discussed)</td>
</tr>
<tr>
<td>Salahi et al. (2020)</td>
<td>2020</td>
<td>Service placement w. eval. envrns.</td>
<td>Partially</td>
<td>Mixed</td>
<td>--</td>
</tr>
<tr>
<td>Gill and Singh (2021)</td>
<td>2021</td>
<td>Simulation frameworks</td>
<td>Partially</td>
<td>User, network</td>
<td>--</td>
</tr>
<tr>
<td>Islam et al. (2021)</td>
<td>2021</td>
<td>res. scheduling w. context classification</td>
<td>No</td>
<td>User</td>
<td>--</td>
</tr>
<tr>
<td>Martinez et al. (2021)</td>
<td>2021</td>
<td>net. design, res. mgmt., eval. envrns.</td>
<td>Partially</td>
<td>Mixed</td>
<td>MEC, cloudlet</td>
</tr>
<tr>
<td>Costa et al. (2023)</td>
<td>2022</td>
<td>Orchestration</td>
<td>No</td>
<td>Mixed</td>
<td>MEC</td>
</tr>
<tr>
<td>Bukhari et al. (2022)</td>
<td>2022</td>
<td>Discovery</td>
<td>No</td>
<td>Mixed</td>
<td>Osmotic comp.</td>
</tr>
<tr>
<td>Songhorabadi et al. (2023)</td>
<td>2023</td>
<td>app. to smart cities</td>
<td>Yes</td>
<td>Mixed</td>
<td>--</td>
</tr>
<tr>
<td>(This paper)</td>
<td>2023</td>
<td>Architectures and algorithms</td>
<td>Yes</td>
<td>Network</td>
<td>(spatial) crowdsourcing, drop comp., osmotic comp., vehicular fog comp., opportunistic fog comp., (MEC and others discussed)</td>
</tr>
</tbody>
</table>

app.: application, arch.: architectures, comp.: computing, eval.: evaluation, net.: network, res.: resource.
applying the concept in custom domains (e.g. automotive, maritime, aviation). Although some of the models seem to overlap, they have sprung from different roots to address disparate objectives, and therefore are considered on an individual basis. Following sections briefly introduce designs identified as models of fog computing concept, along with rationales behind, base scenarios and application environments.

4.1. Osmotic computing

Osmotic computing is a system concept for distributed and federated virtualised environments that aims to automate opportunistic deployment of possibly decomposed applications that span the cloud-to-thing continuum of heterogeneous devices (Villari et al., 2016). The idea behind this form of computing comes form chemistry, where osmosis describes a process of movement of matter from a region of higher concentration to a region of lower concentration. In osmotic computing, osmosis is modelled as a dynamic migration of services to satisfy the infrastructure and application goals that vary over time (Villari et al., 2019). Osmotic computing contributes to fog computing as defined by OpenFog in the areas of scalability and manageability, where it is seen as a distributed application framework for fog environment (Sharma et al., 2018).

Osmotic computing environment is formulated as a regular three-tier system (things–fog–cloud Kaur et al., 2020) composed out of microelements that provide a single abstraction to build data flows that span the whole system. Microelement (MEL) abstraction serves as either a microservice, an information stream, computation unit for a certain class of problems, or a manager that alters or controls the state of a resource. Microelements involved into a dataflow may be deployed to any node in system, including the cloud and things. Fig. 3b presents the idea of osmotic computing in the fog computing context.

Osmotic computing focuses on logical decomposition of the application into a set of cooperating microelements (cf. function as a service concept). Once decomposed, and deployed to virtual components or containers, microelements are subject to aggressive migration. Migration happens between nodes in dimensions constrained by requirements put by the application graph. These include conventional bottom-up offloading from things to a fog, reversed offloading from a cloud to things (Villari et al., 2016), and nodal collaboration. Although the available simulation frameworks for osmotic computing utilise prior global knowledge on network elements and topology (Alwasef et al., 2021), osmotic computing does not put restrictions on decentralised decision-making, and operation in dynamic ad-hoc mobile environments.

4.2. Drop computing

Drop computing is a concept of decentralised computing introduced in Ciobanu et al. (2019b), and later refined in Ciobanu et al. (2019a,c). It is founded on ad-hoc dynamic collaboration of mobile devices enhanced by a human input (social component). Drop computing aims to provide computation and data offloading services in the highly mobile cloud-to-things continuum. That is, computing nodes and service requesters are assumed to be extremely mobile and heterogeneous. Multi-hop opportunistic communication with relay nodes constitute its mobile ad-hoc networking skeleton.

Drop computing adds social characteristics to fog computing concept. Capabilities of a mobile device are extended by the resources available in the crowd of neighbour devices. Furthermore, direct communication between things takes into account their social familiarity to determine the level of detail for exchanged information. A social graph is either introduced by a human factor, or deduced by the devices independently using trust mechanisms.

Applications in drop computing are partitioned into fine-grained tasks. The offloading process starts with the distribution of tasks to nearby nodes. Participation in offloading is voluntary and rewarded. Nodes are allowed to autonomously re-distribute the assigned tasks to the ad-hoc social clusters (communities). Duplication of offloading requests is used to increase the offloading success rate. Based on the availability of different types of computing architectures and the complexity of the tasks, drop computing executes offloading processes either within the crowdsourcing-like system without access to infrastructure nodes, by means of a cloud, or by utilising all the available computing networks. In case of insufficient resources, drop computing routes the offloading request to cloud computing data centres (fall-back). Fig. 3c visualises social relationships between nodes, and task offloading characteristics in drop computing.

4.3. Crowdsourcing

Crowdsourcing forms a system to offload the tasks to a group of heterogeneous worker nodes (Brabham, 2008). The offloading process is managed by a centralised node located in a cloud. Such a cloud-based crowdsourcing service provider discovers the workers, executes task placement and migration, handles incentives mechanisms, and brokers all the bidirectional interactions between the service requester and workers (Jiang et al., 2018). Broker server may be modelled as a distributed system, and, in particular, as a fog computing instance (Yang et al., 2017). Fig. 3e visualises the main characteristics of a crowdsourcing system.

Although crowdsourcing systems tend to be centrally managed, their worker nodes are given a high degree of autonomy. Workers make decisions on task selection and execution through direct interaction with the broker, while indirectly cooperating to build a workflow (Ni et al., 2017). A Set of the accepted tasks specifies a worker’s functional role in the system. Workers can be focused on gathering of the sensor data, processing of the collected sensor data, or both. Worker nodes can interact with a human to consume the human input, or execute some trivial tasks (e.g. yes-no questions, etc.). Processing of the sensors’ data can be delegated to a broker, and then offloaded to the cloud data centres. In a crowdsourcing system, a service requester can actively build an ad-hoc network from the willing to participate worker nodes available in its vicinity.

The spatial form of crowdsourcing extends its capabilities with a location awareness property of offloading requests. In such a system, computing nodes are required to be available at certain physical locations to take part in computation offloading (Kazemi and Shahabi, 2012). A spatial crowdsourcing system is able to instruct a subset of worker nodes to physically travel to particular locations in order to perform the requested tasks. Locations of mobile workers are observed by a centralised broker node as streams of position changes which, in turn, enable the prediction of workers’ location from the collected historical data.

4.4. Opportunistic fog computing

Opportunistic fog computing contributes to mobility-awareness aspects of fog networks in which fog nodes operate in hazardous environments of high density, where resource availability, mobility, and connectivity change dynamically and rapidly (Silva et al., 2017; Fernando et al., 2019). Opportunistic fog computing utilises efficient network scaling to cope with generally unpredictable fog node lifetime. So characteristic for this model of fog computing, a highly adaptive deployment with limited access to infrastructure and cloud data centres, benefits from the dynamic context-aware selection of offloading strategies driven by the properties of the offloaded application.

In response to a service requester mobility, varying network context, and offloading statistics, the peer-to-peer cooperation between opportunistic fog nodes enables dynamic clustering, as well as further load balancing and task migration (Brogi et al., 2020). A dynamic cluster is locally managed by a fog node of transferable cluster head role (Baker et al., 1984). Both the structure and functional properties of a dynamic cluster vary over time. Fig. 3f visualises the key characteristics of opportunistic computing.

6
Fig. 3. Osmotic computing (b), drop computing (c), opportunistic fog computing (f), vehicular fog computing (d), and crowdsourcing (e) as models of fog computing that contribute to its pillars (a).
4.5. Vehicular fog computing

An increasing computational power offered by the on-board units installed in vehicles, and the opportunity of a spontaneous setup of the wireless ad-hoc networks between them (Chlamtac et al., 2003), gave rise to several vehicular computing paradigms (Kai et al., 2016; Fadhil and Sarhan, 2020). Depending on the presence of infrastructure network nodes (e.g. road side units (RSUs), etc.), some authors differentiate between the vehicular edge, and the vehicular fog computing (Hou et al., 2016), while others use both terms interchangeably. The networking foundation highly influences the key characteristics of those paradigms. Certain mobility perspectives can be implicitly excluded from further consideration (Reis et al., 2017; Pereira et al., 2019), while cooperation in offloading and migration processes restricted (Liu et al., 2020; Ning et al., 2019). Moreover, due to network scalability reasons (Kosch et al., 2006), vehicular nodes are usually organised into clusters (Kuklinski and Wolny, 2009), where an elected cluster head manages the cluster members, and brokers the broadcast-based communication.

Albeit it is desirable because of simplification in mobility predictions, vehicular fog computing does not have to be limited to road vehicles (Marín-Tordera et al., 2017; Zeadally et al., 2020). For example, sparsely distributed mobile service requesters can execute cooperative
bring the subject of each reviewed article closer, and the three novel results in few or single representatives works. For certain topics, the literature coverage is considerably low which answer the research questions presented in Table 3 using a method classification provided by its authors. The review process aims to system aspects they contribute to. The applied model is inferred from perspectives: models of fog computing applied by the authors, and the using a unified terminology, and analysed from the two major per-
cal, research works devoted to fog computing deployed over networks that span network layers of multiple heterogeneous devices. Each of the described solutions focuses on particular processes within solutions that span network layers of multiple heterogeneous devices. how node mobility can be addressed to build the complete computing to come up. Five mature models discussed in this section demonstrate
improvements (Madan et al., 2020; Liao et al., 2021; Mohamed et al., 2017) introduced by fog-capable flying ad-hoc networks of unmanned aerial vehicles of strictly defined moving dynamics and paths (Motlagh et al., 2019; Zhu et al., 2021b). Furthermore, aerial, ground, and marine vehicles are expected to cooperate and collaborate to provide a robust fog computing environment (Chen et al., 2020). Vehicular fog computing, pictured in Fig. 3d, constitutes a fertile soil for connecting various ideas, and holds great promise for implementations in the near term. 4.6. Summary

This section presented a selection of the mobility-aware models of fog computing. The authors are well aware of numerous computing architectures that either laid the foundation for fog computing (Satyanarayan, 2001; Balan et al., 2002; Puliafito et al., 2019) or loosely fall under the IEEE 1934 concept (e.g. dew Cristescu et al., 2019, mist Preden et al., 2015; Wang et al., 2019, social dispersed Conway et al., 2019; Garcia-Valls et al., 2018; Yang et al., 2021, or moisture Tufail et al., 2021 computing, etc.) Variuos computing paradigms matching OpenFog’s concept were devised, and the new ones are likely to come up. Five mature models discussed in this section demonstrate how node mobility can be addressed to build the complete computing solutions that span network layers of multiple heterogeneous devices. Each of the described solutions focuses on particular processes within a specific networking environment. Although vehicular fog computing seems to be the most evolved paradigm, the features from the remaining paradigms are required to assure all the benefits of fog computing. Table 2 summarises the characteristics of the discussed five models.

5. Examined research efforts

In this section, a detailed insight into the existing, largely theoretical, research works devoted to fog computing deployed over networks of mobile nodes is presented. The collected publications are discussed using a unified terminology, and analysed from the two major perspectives: models of fog computing applied by the authors, and the system aspects they contribute to. The applied model is inferred from contents of a research paper, although it usually matches the original classification provided by its authors. The review process aims to answer the research questions presented in Table 3 using a method described in Section 5.6.1. Papers are grouped by topics they address. For certain topics, the literature coverage is considerably low which results in few or single representatives works. Research questions in Table 3 enclose the rudimentary ones that bring the subject of each reviewed article closer, and the three novel categories that focus on: multi-tier cooperation within the network, autonomic collaborative behaviour with functional safety aspects, and decentralised application handling during the fog operation. While rudimentary criteria are likely to be required and present in most of the surveys, the remaining three categories are unique to this survey.

A common terminology that follows OpenFog (IEEE, 1934) is established. Discussion on the contents of the reviewed works adheres to the introduced vocabulary. An entity that provides resources for either computation or storage task execution is termed as a fog node, while the entity that issues such a task is known as a service requester. Fog node is labelled as an infrastructure fog node if it does not change its position, otherwise it is referred to as a mobile fog node. Fog nodes may execute tasks originated from service requesters and other fog nodes. In general, offloaders run tasks of the offloaders. A structure that organises fog nodes is a fog network. Tasks are allocated to fog nodes in the fog network to provide fog computing services to the service requesters. Task allocation requires prior resource provisioning to reserve the computational and memory resources within the fog nodes for use by the service requesters (Martinez et al., 2021).

5.1. Osmotic computing

5.1.1. Network function virtualisation (NFV) in the vehicular environment

Authors of Mouradian et al. (2019) propose a solution to a tree-structured application placement onto a network of a hybrid fog–cloud architecture. Applications are modelled as graphs of heterogeneous application components interacting through different types of substructures (e.g. execution ordering relations, loops, conditional selection). NFV framework is utilised as a base for the deployment of the designed algorithms, thus the application placement problem comes down to the mapping of a virtual network function forwarding graph (VNF-FG) onto NFV infrastructure. Allocation of tasks to both tiers of the cloud and mobile fog nodes, as well as the iterative migration decisions, are centralised in the orchestrator functionality. The solution runs a heuristic that relies on aggressive reassignment of tasks until reaching a near-optimal placement, but the impact of migration and its details are not covered. Although placement jointly targets two tiers of a network, the evaluation considers scenarios with a predominant share of cloud nodes. Proposed a priori optimisation depends on the fog node location (estimated from the synthetic mobility model), and a fixed weighting parameter to control the target tier. A single fog node may host multiple application components at a time, however, applications do not arrive dynamically to the system. Replicas are not considered. Both the assumed lower bounds on fog-to-fog link bandwidth capacities, and the maximum task sizes seem unrealistic in practice.
A cloud-to-thing continuum architecture proposed in Sun et al. (2020b) relies on the contract-based resource-sharing mechanisms established among dynamic fog clusters. Clusters are classified by functional domains of IoT applications possible to be handled by cluster members. Cluster heads are nominated by the centralised scheduling algorithm based on their betweenness centrality metric, computing performance, and communication delay to service requesters. Heads subsequently run aggregation of other fog nodes through application of spectral clustering methods. Fog nodes of low mobility are favoured as cluster heads. In the proposed resource-sharing auction, a cloud data centre acts as a trusted third-party, and is engaged in the centralised global task scheduling. The proposed scheduling algorithm aims to overcome the negative impact of fog node mobility by combining the dynamic construction of functional domains with resource sharing among fog clusters. Modelling of fog node mobility is limited to a fog cluster leave-join events. Authors assume fixed high-capacity (1Gbps) links between end devices and fog clusters, and a single cluster to logically cover all the end devices in a city. Migration through nodal collaboration, task decomposition, and reliability are not discussed thoroughly.

## 5.1.3. Collaborative execution of a workflow graph

A mobile fog environment with fog clusters managed by a distributed fog network controller is proposed in Alam et al. (2019). A controller is an autonomic offloading reinforcement learning-based agent that makes local decisions based on the observations of the network state, resource demand and availability. Agents collaborate directly to offload the application parts (workflow of “code blocks”) to either the cluster members or cloud data centres. An offloading decision is consulted with neighbouring offloading agents to minimise the latency. Initially, a mobile service requester runs the application locally to collect the run time statistics. Collected data drives the application decomposition and flow graph analysis. That way, the inter-dependencies and estimated resource consumption are determined. Authors leverage existing cellular networks to i.a. run load balancing between fog networks, but details are not provided. Neither the performance footprint of statistics gathering, nor mobility simulation details are presented.

### 5.1.4. Allocation of sub-tasks with linear relationships

Research in Kimovski et al. (2021) discusses application placement of a graph of optionally sequentially inter-dependent sub-tasks onto virtualised resources of heterogeneous nodes in the cloud-to-thing continuum. The authors describe node mobility with Markov model to reduce the problem space of the meta-heuristic used to solve the multi-objective optimisation problem. State changes of a mobile node are observed at regular time intervals, and modelled as handovers between higher-level network gateways. Real mobility traces are used...
The table below lists surveyed publications with their respective task settings and the corresponding performance metrics evaluated.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhu et al. (2019)</td>
<td>CS Privacy</td>
<td>MDP</td>
<td>DRL</td>
<td>Vehicular</td>
<td>1:1</td>
<td>SUMO, other</td>
<td>T(^5)</td>
<td>R</td>
<td>C</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Sun et al. (2018)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>liu et al. (2021)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Xu et al. (2021)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Zheng et al. (2019)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Xu et al. (2021)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Zheng et al. (2019)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Sun et al. (2018)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Li et al. (2021)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Zheng et al. (2019)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Sun et al. (2018)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Li et al. (2021)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
<tr>
<td>Zheng et al. (2019)</td>
<td>VFC Offloading</td>
<td>Offloading success</td>
<td>–</td>
<td>FedSGD, AAE</td>
<td>Urban</td>
<td>1:1</td>
<td>Other</td>
<td>M</td>
<td>N</td>
<td>★</td>
<td>★</td>
<td>★</td>
<td>★</td>
</tr>
</tbody>
</table>

(continued on next page)
interacting components is provided. Authors constrain optimisation problem to nodes of low mobility, and rely on fault tolerance in case of erroneous mobility prediction on the service requester. Moreover, fog nodes are not autonomic, and the task migration is not discussed thoroughly.

5.2. Drop computing

5.2.1. Offloading in a hybrid social-aware environment

Research in Ciobanu et al. (2019a) strives to jointly minimise the computation offloading completion time, resource and energy consumption in the scenarios of different network architecture setups, where multi-hop opportunistic networking is engaged to enhance the cloud–fog cooperation. Mobile nodes are clustered by social communities they belong to. Social network connections provided by an external systems are explored in device-to-device (D2D) offloading, and in the next-hop selection criteria of routing. Mobility of nodes is simulated with custom mobility traces, and by application of synthetic home-cell mobility model (HCMM) that approximates the human behaviour. Applications for offloading are meant to be split into multiple tasks with strictly defined real-time characteristics, albeit the corresponding impact on the system is discussed not thoroughly. Mobile computing nodes are restricted to smartphones, while more powerful units (termed “fog nodes”) are of fixed positions and underutilised. Both the intra-cluster collaboration, and accidental task redundancy resulting from task dissemination are not explored by the authors.

5.2.2. Strategies for D2D task scheduling

Migration of workload in opportunistic network of mobile nodes with externally-provided social crowd characteristics, and real-time
5.3. Crowdsourcing

5.3.1. Data collection in urban scenario

Authors of Wang et al. (2021a) propose a system in which a central server delegates the deadline-aware sensor data collection tasks to the recruited mobile nodes (“mobile gateways”). Nodes gather the task-specific sensor data from the fixed-position sensors in their proximity. Devised heuristic chooses the most suitable gateways for given tasks, and reorganises the sequence of sensors to be visited in order to minimise the task completion cost (i.e. reward paid to mobile nodes). Authors utilise real human mobility traces. Calculated time slot-based contact probabilities between mobile nodes and sensors estimate the chances of mobile nodes to collect the task-required sensor data within the deadlines. Whilst passively performing as data mules, mobile nodes exhibit their preferences by rejecting the recruitment offer, however they do not collaborate in data collection. Nodes with stable cloud connection are favoured.

5.3.2. Context-aware task allocation

A centrally managed system in which vehicle nodes gather and process visual data collected by other vehicles within their coverage area is proposed in Zhu et al. (2021a). A central coordinator asks vehicles to either video capture fixed-position targets of interest or process the collected video frames from the other vehicles in one-hop distance. A centralised context-aware task allocation scheme is executed per service zone (in evaluation: a single cell) to learn strategies which increase the amount of useful information extracted from the collected data while minimising the processing latency. Proposed task allocation uses complete information of the environment to determine the data collection rate for the sensing vehicles, and assign the processing tasks based on estimated quality of information (QoI) and workload. Authors model fog nodes (data processors) with buses of predefined trajectories, and use real-world mobility data sets in evaluation. Speed and movement direction of the collector vehicles, and the overall number of vehicles are assumed to be constant during a fixed-length decision epoch.

5.3.3. Incentives, and trust management

Sun et al. (2020a) discusses a fog-based privacy protection of the personal data in the bidding process within the designed incentive mechanism. Authors aim to reduce system response time and communication overhead. Collaborating buses are organised into one-hop-range fog clusters (“regions”) that are maintained through a direct communication and broadcasting. Clustering is driven by the temporal information derived from the historical locations of buses. Clusters are concentrated around an area of a high bus traffic, and remain valid until the next bus route planning details update. Vehicles serve as smart sensor platforms, data collectors, or assist in incentive mechanisms. Reward data is routed between clusters in opportunistic manner. Fog nodes proxy task assignment to worker vehicles.

In Wang et al. (2020b), a spatial crowdsourcing setup embedded in a mobile fog computing environment with worker fog nodes visiting the fixed-position heads of sensor clusters forms a test bed for evaluation of a trustworthy data collection with fog node path planning algorithm. Algorithm minimises energy consumption and transmission delay of the task execution. Fog nodes apply direct and indirect metrics to estimate the sensor’ trust values and levels, so that only data from trustworthy sensors is gathered. Direct trust metrics exploit historical communication interactions of the sensor node, while the indirect ones combine trust information from the neighbours. Trust values drive the cluster head election in stable sensor networks, and guide the fog node mobility. Fog nodes do not collaborate with each other, and have limited moving distance.

5.3.4. Community detection to build an overlay network

Beli et al. (2020) extends existing infrastructure networks with mobile nodes elected from candidate mobile user devices by means of periodic application of centralised evolutionary community detection algorithms. Part of the fixed-position nodes take roles of crowdsourcing brokers, while some of the mobile nodes opportunistically collect data from workers and proxy access to the brokers in a cloud. Authors measure effectiveness of the selected community detection algorithms in the identification of evolving communities. Measurements of the members’ centrality metrics drive the designed mobile node selection strategy. Mobility is simulated with real human mobility traces, however application to a real-world use case is not discussed.

5.4. Opportunistic fog computing

5.4.1. Sequential task graph execution

In Wang et al. (2021b), service requesters located in the resource-constrained vehicles offload the structured applications to the selected nearby resource-rich vehicles. An application forms a graph of tasks bound by a data flow, where a task is either run locally or offloaded from a vehicle to a vehicle. Offloadee selection, task allocation, and data transfer happen on a regular basis, however the assigned tasks always run to completion with neither migration nor fault management involved. Evaluation is limited to a single application in a single-tenant environment. Synthetic microscopic mobility model simulates vehicle moves along a unidirectional straight road of a predefined length.

5.4.2. Distributed situation recognition

Audrito et al. (2021) tackles a problem of coordination of the dynamic IoT systems using the distributed centrality measures applied to the incomplete time-varying network graphs with injected faults. Authors use properties of direct neighbourhoods to establish a classification of arbitrary large dynamic networks that preserves locality of interactions. Proposed algorithms are used to nominate local coordinators to support the use cases of situation recognition tasks like network vulnerabilities detection, and election of cluster heads of “fog colonies”. Centrality measures are calculated by nodes locally and asynchronously. Synthetic mobility with Lévy walks is modelled. Work lacks extensive evaluation in practical setups.

5.4.3. Task replication, and node redundancy

Signalised road intersection scenario and task replication are investigated in Yang et al. (2020) to enhance performance of deadline-aware task offloading. A fixed budget of task copies controls the redundancy level. No recovery mechanisms for partial offloading results are discussed. Offloaders receive an offloadee candidate list at intersections. Joint consideration of traffic data (e.g. traffic lights state) and vehicle movement is exploited to predict the future vehicle position. Although vehicles follow realistic mobility patterns, the environment is semi-static.

In Asensio et al. (2020), a fault-tolerant multi-agent hierarchical system of fog clusters with dynamically elected cluster heads’ backup nodes (“leaders”) guarantees service continuity, and increases the architecture robustness. Once the current head fails or suddenly leaves the network, a backup node retains the access to cloud management.
functions and a copy of the storage. Leaders are managed by a cloud-based centralised agent, whilst they coordinate the respective cluster members. Centralised agent runs heuristics to derive a set of actions to map the nodes from a physical to a logical topology through iterative members. Centralised agent runs heuristics to derive a set of actions to map the nodes from a physical to a logical topology through iterative connectivity-based k-means-based clustering and node role assignment strategies. Cluster heads apply those actions in a decentralised manner. Authors minimise a weighted-sum objective, where mobility of a node is a weighted discrete variable. There is no further evaluation with real-world traces mobility.

5.4.4. Container placement and migration

Mseddi et al. (2019) discusses containerised task offloading to a clustered fog network of nodes classified by their mobility patterns. A cluster head ("domain controller") executes an online resource allocation strategy that deploys containers to cluster members (or fallbacks to a cloud), so that the number of satisfied offloading requests is maximised. Workload migration between nodes happen if the continuously monitored fog-requester communication and processing latencies exceed certain thresholds. Nodes are categorised by their mobility degree (immobile, slow-moving, fast-moving), and evaluate the solution with (respectively to the category) real mobility traces. However, cluster head mobility is not discussed thoroughly, and the evaluation considers only a single cluster (neither cluster formation, maintenance, nor cooperation are taken into account).

5.4.5. Cooperative sensing in a platoon

Du et al. (2020) proposes an infrastructure-less vehicular system with clustered adjacent vehicles sharing their sensing, communication, and computation resources to enhance the safety of autonomous driving. Platoon-head vehicle allocates visual sensing tasks to the vehicles of heterogeneous sensing capabilities for cooperative processing in order to maximise the non-overlapping sensing coverage. The head vehicle fuses the task results to get an accurate state of the platoon, and to feed the distributed deep learning-based lane change manoeuvre prediction algorithm. The prediction is limited to mandatory and discretionary situations, and vehicle selection lacks extended context-awareness. Mobility is simulated with real-world traces, however, vehicles are driving at a relatively stable speed and distance (platoon formation condition).

5.4.6. Collaborative task offloading

Buda et al. (2020) formulates a two-stage algorithm for non-monolithic task offloading in a clustered vehicular environment to one- and two-hop offloadees. The first stage leads to a cluster formation, while in the second stage a cluster head either partitions and allocates tasks to members or requests a collaborative processing from neighbouring clusters. Tasks are executed in parallel to boost completion rate. A decentralised fuzzy logic-based heuristic for cluster head election consumes data exchanged between one-hop neighbours, while fuzzy membership functions and offloadee selection take advantage of expert knowledge and context-aware metrics. Mobility is simulated with a freeway model (Wu et al., 2013). Details of collaboration between cluster heads are not covered, while the task and result data sizes are of fixed small values during evaluation.

5.4.7. Temporal-aspect-driven task offloading

In Zhu et al. (2019a), a distributed online offloading scheme ("Chameleon") of image-based object recognition monolithic tasks from vehicular requesters to bus fog nodes aims to process data of higher image quality with lower latency. A spatio-temporal variation in vehicular traffic density is explored to define time periods ("buckets") in which traffic density changes regularly. Workload of a fog node is expected to evolve as a Markov chain within a time bucket of equally-sized time slots. At the beginning of a time bucket, service requester selects an offloadee from the one-hop candidates. Image resolution is adjusted to the workload observations. Evaluation with a visual-based assisted driving application considers both the infrastructure and mobile nodes. Simulation uses custom mobility traces extracted from real data. Neither nodal collaboration, extended context-awareness, nor highly congested situations are addressed in detail.

5.4.8. Social-aware data offloading

Envisioned in Radenkovic et al. (2018), a context-aware peer-to-peer collaborative content sharing scheme applies predictive heuristics to overlay networks of connected infrastructure and mobile nodes in dynamic disconnection-tolerant networks. Overlay networks encode social topologies with relationships between the content publishers and subscribers, and nodal temporal perspectives ("ego networks") to model the nodes met. Nodes observe and exchange upon a contact the time-varying local content popularity. Social metrics (e.g. betweenness, centrality) of neighbours are examined to manage heterogeneous chunks of content with individual popularity, and adaptively forward caching requests based on fused local popularity. Nodes run neighbour offloading of least popular content to prevent from resource exhaustion. Although mobility of nodes is simulated with real-world data sets, the solution lacks extended evaluation in real environments.

5.5. Vehicular fog computing

5.5.1. Cooperative offloading

Delegation of tasks to vehicles for cooperative execution can either support the overloaded RSUs (Ye et al., 2016; Lv et al., 2021), or reduce the negative impact of the mobility-induced migrations on task processing (Lv et al., 2021), or optimise resource utilisation in offloading (Zhou et al., 2020; Tan and Hu, 2018). In particular, Lv et al. (2021) enables the vehicles that are passing through two adjacent RSUs’ handover area to apply the learnt optimal one-hop offloading schemes, in Ye et al. (2016) bus fog nodes of known positions extend RSU capabilities in incentive-aware task processing, Zhou et al. (2020) splits offloading into two phases: request gathering and issuing of scheduling policy to be locally executed at each fog node, and final fog node selection based on received policies run by a requester, Tan and Hu (2018) discusses coded content caching in short and long time horizons ("timescales"), where the Q-learning model optimises cost of resource allocation.

Vehicles of public transport mobility characteristics, like taxis (Lv et al., 2021) and buses (Ye et al., 2016) are likely to become offloadees in cooperative offloading due to their predictable mobility trajectories. Candidate vehicles are selected either centrally or as RSU: based on the predicted trajectories (Lv et al., 2021), and accurate positions observed in real-time (Ye et al., 2016; Tan and Hu, 2018), or distributively: by requesters, based on scheduling policies calculated by neighbouring fog nodes (Zhou et al., 2020). While Ye et al. (2016) solely nominates vehicles as offloadees, Lv et al. (2021), Zhou et al. (2020) and Tan and Hu (2018) involve infrastructure and mobile nodes as candidates. A fixed radius (Lv et al., 2021; Ye et al., 2016), or a communication range is explored (Zhou et al., 2020; Tan and Hu, 2018) in candidate discovery. Although (Lv et al., 2021) mentions task segmentation to execute partial offloading, no further discussion follows. Coded caching in Tan and Hu (2018) supports costly download of missed data segments from infrastructure nodes, while remaining works deal with monolithic tasks. Neither task redundancy nor multi-tenancy (results from prevailing binary ILP formulation of task assignment) are present. Works do not model the offloadee behaviour (e.g. no task rejection policies, selfishness, etc.), and no autonomous collaboration is found. Dynamic task arrival is rarely explored. Solutions lack evaluation in realistic networking setups.
5.5.2. Caching with federated learning

A self-driving environment, where cellular base stations provide arrival times of vehicles at each RSU is given in Yu et al. (2021) as a background for discussion on tasks offloading of monolithic model training to subsets of neighbour vehicles selected for participation in the federated learning-based content popularity prediction. Vehicles learn the AAE shared model with their private data of diverse quality. Global model at RSU is repeatedly updated by aggregating the locally trained ones. Contents cached in RSUs are cooperatively replaced in assistance of a base station that populates subsequent RSUs with the predicted popular contents. Population happens in response to movement of a vehicle group along the road with adjacent road site units, however only the RSU located at the entrance to the road runs the popularity prediction. Solutions lacks simulation of vehicular mobility with real-world data sets and reliability modelling. Context-awareness is only exploited for the mobile user in the vehicle.

5.5.3. Clustering based on road topology

In Wu et al. (2021) vehicles in the coverage area of an RSU are organised into directional groups ("fog cells", "subnetworks") according to their turning directions at the next crossing. Groups aid the centralised monolithic task offloading scheme run by other vehicles. Idle resources in vehicles and RSUs are considered jointly. RSU exploits the complete state information of the network it covers to select the offloadees, and execute offloading. Collaboration for task offloading is only allowed with RSU and within the same group. However, a vehicle can process at most one task simultaneously, and RSU-assisted handover of offloading results is only covered in simulation. Vehicles are expected to obey the traffic rules (e.g. sudden lane change is prohibited), and they follow the generated mobility patterns.

5.5.4. Resource pooling

Resource pools in VFC are modelled as clusters managed by either infrastructure (Tang et al., 2021; Sorkhoh et al., 2020; Hameed et al., 2021) or mobile (Cha et al., 2021) cluster heads. In particular, Tang et al. (2021) sketches a system with incentives offered by RSUs to offloadees on behalf of requesters, in Sorkhoh et al. (2020) RSU schedules the execution of tasks with deadlines either at the RSU or at in-range vehicles, while other works deal with optimal formation of on-demand (Cha et al., 2021) or scheduled (Hameed et al., 2021) clusters that execute tasks within a time limit. Cluster is formed either entirely bottom-up voluntarily through interactions of candidate members with the head (Tang et al., 2021), collaboratively through explicit confirmation of the membership (Cha et al., 2021), or exclusively by the head (Sorkhoh et al., 2020; Hameed et al., 2021). While Tang et al. (2021), Cha et al. (2021) and Hameed et al. (2021) cluster limit set of member types to vehicles, Sorkhoh et al. (2020) includes also RSUs. Stability of a cluster depends on its (re-)construction metrics, and determines its effectiveness in execution of the requested tasks. Tang et al. (2021) considers vehicle’s dwell time in the coverage of RSU, available resources, and cluster joining reward in community formation metrics. Cha et al. (2021) and Hameed et al. (2021) exploit contextual information from V2X safety beacons (e.g. position/distance, velocity, direction, etc. – cf. IEEE 802.11p BSM) to recruit the cluster head (in Hameed et al. (2021)) or members (in Cha et al. (2021) and Hameed et al. (2021)). Cha et al. (2021) proposes a “companion time” (i.e. an upper limit of link duration between two nodes) to measure the cluster lifetime. Furthermore, Sorkhoh et al. (2020) operates within a single global cluster. However, in Cha et al. (2021) vehicles of speeds above a fixed limit are ignored (to favour stable clusters), and Hameed et al. (2021) does not support self-nomination: cluster head is selected by an infrastructure node (“fog gateway”). Although (Tang et al., 2021) aims to decentralise decision making process, there are discussed neither collaboration, respective cost analysis, nor confirmed autonomy of vehicular choices. The evaluated there GA-based optimisation assumes fixed number of vehicles and communities (implies centralised or hybrid management).

5.5.5. Task replication

A work in Sun et al. (2021) studies a relation between the number of task replicas and execution delay. The devised two-step approach to the replica number estimation consists of a RSU-centralised rough approximation for a long time horizon (“timescale”) based on statistical network conditions, which is subsequently autonomically fine-tuned by vehicular offloaders for a short time horizon using sequential decisions (CMAB). Goal of an offloader is to choose a subset of offloadees for a replica assignment so that the offloading delay is minimised. Such a hybrid scheme is promising in balancing the management workload while retaining context locality. However, in this work, tasks are strictly homogeneous (incl. deadlines and types) and run to completion (i.e. pending redundant executions are not cancelled), task rejection is not discussed (i.e. it assumed that at least one offloadee accepts the task), no offloadee collaboration happens, cost of local management lacks analysis, replica calculation is not proactive (scheme runs once task becomes a task queue head), and mobility traces are synthetic (generated for a highway scenario).

5.5.6. Multi-period task offloading

Liu et al. (2021) proposes a multi-period offloading which is an extension of Wang et al. (2020a) for hybrid environments. Designed scheme is similar to the two-step approach from Zhou et al. (2020), but with scheduled allocation attempts of the remaining tasks. That is, tasks that are not selected by the offloadee candidates are broadcasted to nodes contacted in the next periods with hope for an acceptance. Simplified incentive mechanism is devised to reward fog nodes in dynamic programming-based decision making. Requesters run a heuristic-based node selection that aims to maximise task completion ratio. There is no task redundancy, reliability aspect is missing (i.e. assumed uninterrupted communication and sufficient dwell time), and the simulated urban mobility scenario lacks details.
5.5.7. Offloading under information incompleteness and uncertainty

Work in Zhou et al. (2019) designs an offloadee selection scheme that combines centralised contract signing and pricing-based iterative matching with local preference learning at the requester. Having a complete information, the operator quantifies resource sharing capabilities of vehicular offloadees (i.e. assigns “types”), while under information asymmetry only estimation of type probability is available. Operator aims to maximise its long-term profit by signing contract items with the willing vehicles, so that certain amount of resources at some price (“matching cost”) is shared by given vehicle types. At the other end, a requester with complete information issues a list with preferred offloadees to the central unit for one-to-one matching with an offloadee. However, once having no global overlook, the requester tries to minimise its long-term offloading delay by adjusting the parameters in the preference calculation (i.e. history of selections and observed delays) with online learning (MAB). Authors consider neither context-awareness in preference construction, task redundancy (incl. handling of multiple tasks at each time slot), offloading fallback details, mobility simulation with real traces in scenarios other than a highway, nor incentive system modelling.

5.5.8. Semantic-aware real-time data processing

Authors of Fu et al. (2021) propose a live video transcoding system in C-V2X environment, where RSU allocates transcoding requests to fog nodes (buses, commercial fleets). Time-varying resources and link quality are modelled as fully observable stochastic processes (of values discretised at finite number of levels) to form a MDP for joint optimisation of bitrate adjustment, offloadee selection (no redundancy), and resource allocation. Proposed scheme is centralised at RSU to run SAC learning. Authors evaluate different categories of video streams (e.g. sport, cartoon, etc.), however each such a segment is of a fixed playback time (cf. task deadlines in Sun et al. (2021)). Semantics of a stream category put specific requirements on transcoding resources’ consumption. Although custom traces are supposed to drive the mobility, the data set lacks details. Serving latency rapidly increases with growing number of requesters, while large (over 8) number of fog buses leads to serious communication issues.

5.5.9. Event-triggered offloading and migration

Zhu et al. (2019b) devises a task allocation scheme (“Folo”) that assigns containerised video streaming and recognition service instances to commercial fleets (e.g. taxis, buses). The existing cellular networks are reused for scheme deployment (base station), communication, node discovery and tracking, and to trigger iterative live migration upon handover events. Additionally, a requester discovers candidate offloadees with 802.11p one-hop broadcasts. Solution maintains a weighted trade-off between the service latency and quality loss (discretised at several levels). Although the complete system information is explored, the scheme suffers from scalability issues during high density of vehicular traffic (simulated with custom traces). Solution exhibits high memory consumption, even though number of video streams is relatively low (up to 8), and simulation time is relatively short (60s).

5.5.10. Improving network connectivity with forwarding and proxying

Apart from task and data offloading, some authors propose solutions to delivery of data sensed by vehicles (Liang et al., 2021), and drain of cellular radio resources (Jabri et al., 2019) within the dynamic environments. In Liang et al. (2021) a fog node rebuilds the data forwarding path at each hop if the current data relay vehicle is likely to leave its communication range. The selected adjacent vehicle serves as a data mule that runs store-carry-forward of as much sensor data as possible towards the processor (“fog node”). Furthermore, Jabri et al. (2019) aggregates cloud-directed traffic by nominating proxies (“gateways”) around which a fixed number of vehicles can cluster. Candidate gateways are self-nominated with fuzzy logic scheme which consumes degree centrality and link quality measurements. After vehicles broadcast self-nomination results to neighbours, the ACO-based velocity-oriented clustering happens. Both reviewed solutions are evaluated with urban scenarios (fog nodes are taxis (Liang et al., 2021) or buses (Liang et al., 2021; Jabri et al., 2019)) of low mobility (30–60 km/h (Liang et al., 2021), 0–20 km/h (Jabri et al., 2019)) using real (Liang et al., 2021), simulator-generated (Liang et al., 2021; Jabri et al., 2019), and synthetic (Jabri et al. 2019) traces. While Jabri et al. (2019) assumes that vehicles never leave the service area, and there is a MEC infrastructure (scheme initiator, service fallback) present, Liang et al. (2021) expects that relays have enough time to store and forward the received remaining data. Both works neither include redundancy (in: gateways, auxiliary relays), load balancing (of: communication overhead, and fog process), nor security threat analysis. Liang et al. (2021) deals well with high density traffic, but suffers from unavailability of candidate relays (i.e. applies active waiting), rapid changes in density, and higher mobility of fog nodes. Jabri et al. (2019) observes a lower ratio of connected vehicles in mobile scenario. Interestingly, Liang et al. (2021) analyses the computational cost of its algorithm.

5.5.11. Overloaded node management

Yadav et al. (2020) delegates mobile users’ tasks from overloaded infrastructure (“cloudlet nodes”) to vehicles of commercial fleets (buses, taxis). Proposed scheme relies on the locally (at a cloudlet) estimated future resource demand. Measurements are collected by a “zone manager” (at a base station) to trigger the task and offloadee selection procedures. Cellular networks’ spatial organisation and facilities are reused (cf. Jabri et al. (2019), Yu et al. (2021) and Zhu et al. (2019b)). Collaboration happens neither between zones nor vehicles, thus the devised heuristic-based scheme suffers from varying traffic density, task storm due to number of requesters, and higher velocities. Mobility is simulated with a synthetic model, and generated traces (urban scenario). Authors provide a computational complexity analysis.

5.5.12. Microservices

Scheme in Sami et al. (2020) allocates containerised service instances (cf. Zhu et al. (2019b) and Cha et al. (2021)) to clusters of vehicles. Specification of a service comprises inter-dependent microservices, each mapped to a fog node in the same cluster. Multi-objective maximisation of the deployed services is solved with a genetic algorithm which repairs the infeasible individuals (through container migration, cluster extension, and change in the service specification). A hybrid multi-layer network architecture is exploited in failure handling (e.g. connectivity, head recovery, migration, etc.), approval of the cluster offers by the requester, and intra-cluster load balancing. A cluster head self-election happens locally (QoS-OLSR). Kubeadm tooling is extensively used in the management of clusters. Evaluation with real data sets excludes the node heterogeneity, while connectivity to the orchestrator (head or RSU) is assumed stable. Authors describe multiple features of the devised system, however, there are almost no in-depth descriptions.

5.5.13. Node redundancy to improve computing reliability

Dong et al. (2020) discusses reliability for task offloading to a cluster of vehicles (“helpers”) with backup nodes (“followers”) that calculate at reduced rates (“shadowing replication”). The designed recovery mechanism substitutes a node upon its failure or cluster leave with the associated replica, speeds up the replica calculation rate, and nomimates a replica for it. Leave notification is issued in advance. Through boost in the rate, the execution states of a task segment assigned to the replica and the faulty node are expected to match. That is possible by setting an optimal lower bound on the calculation rate, which is a trade-off between the energy consumption (grows with higher reduced rates) and latency (affects task completion ratio). Followers do not have to belong the followers’ cluster, however
task halts if they are not present in the head’s range during recovery. Overlapping communities are prohibited (i.e. a helper belongs to a single cluster), there is exactly one replica for a node, task segments have no dependencies. Moreover, neither head recovery nor allocation of functional layer (e.g. security management, results assembly, cluster disbanding, etc.) to the hybrid networking environment are discussed in detail. Evaluation does not provide information on the simulated mobility (highway scenario). The number of segments is medium (10), while the number of fog nodes is an order of magnitude larger than the service requesters.

5.5.14. Modelling the effects of mobility

Saad and Grande (2020) devises MDP-based solution to represent and estimate the link quality (“connectivity status”) of a vehicle communicating over multiple heterogeneous media (e.g. V2X, LTE, etc.) including handovers, simultaneous and interleaving connections. Locally calculated quality data are comparable (e.g. may extend the ranking metrics in a candidate offloadee selection), and remain valid even if the hops are increased (e.g. due to mobility). RSU records the advertised qualities to run vehicle behaviour estimation. Authors do not discuss the long-term estimations of link quality and intermittent connections. Simulation relies on a single-RSU urban environment and generated mobility traces. Similarly to Dong et al. (2020), the main focus is put on a detailed design of a VFC building block, rather than a whole system of partially modelled components.

5.6. Methodology details

This survey examines selected research works in the domain of fog computing with mobile fog nodes. In order to clearly expose the core ideas taken up by the authors, as well as to facilitate the readers’ understanding, the articles were summarised using the developed unified vocabulary. Original wording was additionally given in quotation marks. Next sections outline the details of the adopted methodology, and provide guidance to the presentation of results.

5.6.1. Method

An iterative multi-step method was applied to search for and to qualify the research papers. Although, initially inspired by Kitchenham et al. (2010), the developed selection and qualification methods were loosely based on the presented there process due to the subject specifics resulting in i.a. notably scarce coverage of the matter in publications. The conducted iterative method comprised two stages executed after preliminary abstract screening had filtered out the irrelevant search results.2 In the first stage (initial screening), a careful inspection of the system model as delineated in each article led to matching a fog computing model. Furthermore, the general characteristics of the modelled problem were gathered and tabularised. First stage was completed with two-way snowballing process that identified valuable references to either be directly included in next iterations, or trigger scheduling of citation alerts, or both. Second stage of the method left aside all the papers that did not involve mobile fog nodes as computation entities. Consequently, the detailed reading of the full texts was started for the qualified papers. Peculiarities of the system models were contrasted with the results of evaluations carried out by the authors. In the meantime, authors ideas were probed to match and extend the developed unified vocabulary. Both the temporal (e.g. conference papers predating the corresponding journal publications), and functional (e.g. equivalent contribution) duplicates were identified and resolved. The devised review method is quality-oriented, thus publications with unclear, incomplete or inconsistent contribution were low ranked.

Execution of the review method led to a qualification of 1172 papers in total. Collected fog computing-related works span over years 2013–2021, however those published after 2017 were considered foremost to increase the probability of OpenFog reference and awareness. Iterative application of the first stage cut down the overall number of articles in a pool to 140 entries. Execution of the second stage left 41 papers presented in this survey. New publication alerts provided by the scheduled search queries, and citation alerts were inspected weekly as the subjects of the review method until mid of October, 2021. Scopus abstract and citation database, and the related tools, were used to query and monitor the peer-reviewed literature.

5.6.2. Presentation

Surveyed articles are categorised, and presented from multiple perspectives using various means. Decimal percent values were rounded up to whole numbers in all the figures to increase diagrams’ readability. Table 5 provides a bird’s eye view of the reviewed domain, and the related open issues. Domain of a fog computing concept is divided into its models, where the percentage contribution is expressed as a ratio of a number of articles linked to a particular model to the total number of surveyed articles. Bar plot-visualisation of the selected aspects (Table 3) with coverage in fog computing models is given in Fig. 4. Illustration of mathematical frameworks employed, and optimisation goals set by the authors of the surveyed research works without an distinction into fog computing models is provided in Fig. 5. Table 4 provides a summarised information on the surveyed papers.

6. Open issues and research opportunities

Although formally conceptualised in 2018 with IEEE 1934 as an adoption of the OpenFog Ref. IEEE (1934), the ideas emphasised and incorporated by fog computing were evolving over years. Research studies on providing the computing and storage functionalities to the resource constrained end users led to the emergence of a multitude of overlapping architectures and algorithms to address the selected areas of the problem. Several of those solutions that comply with the fog computing characteristics are considered in this survey under the umbrella of OpenFog formulation. Massively distributed computing environment in which fog is anchored poses numerous challenges on multiple system levels that touch various actors of the system. This survey aims to picture the current state of the academic research on fog computing with mobile fog nodes, where the mobility-awareness is seen as a key enabler for highly scalable adaptive fog solutions. This section outlines the spotted issues, as well as presents the opportunities of further research studies. Table 5 summarises the open points topic-wise.

6.1. Osmotic computing

OC offers rapid on-demand flow of tasks in multiple dimensions within the cloud-to-thing continuum. Works that contribute to OC suffer from the highly centralised management (Figs. 4(a), 4(b), 4(o)). Decentralisation should improve scalability of solutions, enable fine-grained context-aware decisions (cf. Mouradian et al. (2019)), and would look kindly on nodes of higher mobility (cf. Kimovski et al. (2021) and Sun et al. (2020b)). Although migration seems to be the main feature of OC, its evaluation is rarely in-depth (Fig. 4(h)). Papers lack comprehensive analysis of the schemes’ cost and performance footprint (Mouradian et al., 2019; Alam et al., 2019), incorporation of
fault resilience (Sun et al., 2020b; Mouradian et al., 2019), and details of interoperability with existing networking environments (Kimovski et al., 2021; Alam et al., 2019). To improve solutions’ reliability, simulations should run in realistic networking setups with real mobility data sets (Fig. 4(p)).

6.2. Drop computing

DC scales well over hybrid networks through collaborative decisions that take into account the social component. In its current research phase, it is limited to human behaviour, node heterogeneity is low (Figs. 4(c), 4(m)), interoperability is left implicit, and simulator support lacks alternatives (Ciobanu et al., 2019a; Nistor et al., 2021). Although DC heavily relies on opportunistic networking, neither redundancy nor hybrid management and hybrid collaboration are eagerly explored (Figs. 4(j), 4(k), 4(n)). More work is expected in exploitation and alteration of social graphs (both the locally-maintained and externally-provided) that result from the context-aware decisions.

6.3. Crowdsourcing

CS services connect its users with remote sensing units (devices and human beings). Current incarnations of CS focus on addressing the topics in centralised management and interfaces towards the users. Local aspects of the worker-side part of the system have no decent coverage yet. CS favours nodes of stable connectivity (Wang et al., 2021a), highly predictable trajectories (Zhu et al., 2021a; Sun et al., 2020a; Wang et al., 2020b), and accurate mobility patterns (Wang et al., 2021a; Belli et al., 2020). Even if a clustered environment is considered (Sun et al., 2020a; Wang et al., 2020b; Belli et al., 2020), cluster structures are either fixed (Sun et al., 2020a) or maintained globally (Belli et al., 2020), and cluster collaboration is rare (Sun et al., 2020a). Due to sparse interactions between workers (Sun et al.,
2020a; Wang et al., 2020b; Belli et al., 2020), the distributed context-awareness is not satisfactory (Figs. 4(d)–4(i)). Therefore, scalability seems to be an afterthought in CS.

6.4. Opportunistic fog computing

OFC brings promise of successful offloading within the highly dynamic infrastructure-less networks (e.g. V2V network of vehicles). Although there exist several attempts to formulate complete solutions, they fall short in many ways. OFC relies on opportunistic networking, thus sophisticated handling of context-awareness, proper fault management (Audrito et al., 2021; Asensio et al., 2020), and extensive evaluation with real data sets and realistic networking conditions are expected, but rarely addressed simultaneously. Furthermore, available OFC solutions lack modelling of the extended node behaviour (e.g. for complex situations understanding (Zhu et al., 2019a), collaborative verification of observations (Radenkovic et al., 2018), functional specialisation (Fig. 4(m)), migration (Yang et al., 2020), etc.) that runs within the possibly hybrid environments (Asensio et al., 2020; Msoodzi et al., 2019).

6.5. Vehicular fog computing

VFC explores the potential of the hybrid environment of VANETs and the hierarchical infrastructure. VFC scenarios require efficient scalability mechanisms to assure satisfying RAS for their services. Existing solutions rarely model (Fig. 4(l)) and explore the multi-dimensional collaboration thoroughly (as for e.g. migration (Zhu et al., 2019b; Lv et al., 2021), results’ handover (Wu et al., 2021), control plane redundancy (Dong et al., 2020), incentives (Tang et al., 2021), security, etc.), leaving the long-term service continuity aspect uncovered. Works tend to model the complete systems of implicit features, rather than focus on design and extensive evaluation of particular components (cf. Dong et al. (2020) and Saad and Grande (2020)). In their current shape, VFC systems are not inclusive enough: fog nodes are limited to certain classes of vehicles of implied behaviour (e.g. buses, taxis, etc.) analysed in the preconfigured environments.

6.6. Synergistic outlook: lessons learnt

To further interpret the review results, we treat the presented fog models as building bricks of a working fog solution. That is, they are neither utterly orthogonal nor undeniably overlap altogether. The underlying fog foundation promotes desirable scalability, and this is the aspect in which most of the presented papers do not meet the
Table 5
Topics addressed by the surveyed papers and their percentage share in total.

<table>
<thead>
<tr>
<th>M. Share</th>
<th>Topic</th>
<th>Refs.</th>
<th>Open points</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC 10%</td>
<td>Network function virtualisation (NFV) in the vehicular environment</td>
<td>Mouradian et al. (2019)</td>
<td>Context-aware offloadee selection/migration/case study/eval. w. rwmds.a, pne.b</td>
</tr>
<tr>
<td></td>
<td>Resource auction among fog clusters</td>
<td>Sun et al. (2020b)</td>
<td>Service reliability/migration/scalability/eval. w. rwmds.a, heterogeneous networking environs.</td>
</tr>
<tr>
<td></td>
<td>Collaborative execution of a workflow graph</td>
<td>Alam et al. (2019)</td>
<td>Offloader cost analysis/load balancing and sharing/explicit node mobility/eval. w. mobility data.</td>
</tr>
<tr>
<td></td>
<td>Allocation of sub-tasks with linear relationships</td>
<td>Kimovski et al. (2021)</td>
<td>Service reliability/scalability/migration</td>
</tr>
<tr>
<td>DC 5%</td>
<td>Offloading in a hybrid social-aware environment</td>
<td>Ciobanu et al. (2019a)</td>
<td>Heterogeneity/unbounded collaboration/context-aware offloadee selection</td>
</tr>
<tr>
<td></td>
<td>Strategies for D2D task scheduling</td>
<td>Nistor et al. (2021)</td>
<td>Scalability/eval. w. rwmds.a</td>
</tr>
<tr>
<td>CS 12%</td>
<td>Data collection in urban scenario</td>
<td>Wang et al. (2021a)</td>
<td>Collaboration/node heterogeneity/sensor mobility</td>
</tr>
<tr>
<td></td>
<td>Context-aware task allocation</td>
<td>Zhu et al. (2021a)</td>
<td>Scalability/sensor mobility/eval. w. pne.a</td>
</tr>
<tr>
<td></td>
<td>Incentives, and trust management</td>
<td>Sun et al. (2020a)</td>
<td>Heterogeneity/sensor mobility/eval. w. rwmds.a</td>
</tr>
<tr>
<td></td>
<td>Community detection to build an overlay network</td>
<td>Belli et al. (2020)</td>
<td>Scalability/case study/eval. w. pne.b</td>
</tr>
<tr>
<td>OFC 22%</td>
<td>Sequential task graph execution</td>
<td>Wang et al. (2021b)</td>
<td>Scalability/migration/eval. w. rwmds.a</td>
</tr>
<tr>
<td></td>
<td>Distributed situation recognition</td>
<td>Audrito et al. (2021)</td>
<td>Scalability/case study/eval. w. rwmds.a, pne.b</td>
</tr>
<tr>
<td></td>
<td>Task replication, and node redundancy</td>
<td>Yang et al. (2020)</td>
<td>Scalability/recovery mechanisms/eval. w. rwmds.a, pne.b</td>
</tr>
<tr>
<td></td>
<td>Clustering based on road topology</td>
<td>and Asensio et al. (2020)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Collaborative task offloading</td>
<td>Mseddi et al. (2019)</td>
<td>Cluster cooperation/eval. w. pne.b</td>
</tr>
<tr>
<td></td>
<td>Cooperative sensing in a platoon</td>
<td>Du et al. (2020)</td>
<td>Context-aware offloadee selection/eval. w. pne.b</td>
</tr>
<tr>
<td></td>
<td>Temporal-aspect-driven task offloading</td>
<td>Zhu et al. (2019a)</td>
<td>Case study/eval. w. rwmds.a, pne.b</td>
</tr>
<tr>
<td></td>
<td>Social-aware data offloading</td>
<td>Radenovic et al. (2018)</td>
<td>collab. w. context-awareness/scalability/eval. w. pne.b</td>
</tr>
<tr>
<td></td>
<td>Container placement and migration</td>
<td></td>
<td>Extended node behaviour/eval. w. pne.b</td>
</tr>
<tr>
<td></td>
<td>Resource pooling</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cooperative offloading</td>
<td>Lv et al. (2021), Ye et al. (2016), Zhou et al. (2020) and Tan and Hu (2018)</td>
<td>Collaboration/reliability/eval. w. rwmds.a, pne.b</td>
</tr>
<tr>
<td></td>
<td>Caching with federated learning</td>
<td>Yu et al. (2021)</td>
<td>Reliability/collaboration/eval. w. rwmds.a</td>
</tr>
<tr>
<td></td>
<td>Clustering based on road topology</td>
<td>Wu et al. (2021)</td>
<td>Cluster cooperation/eval. w. rwmds.a, pne.b</td>
</tr>
<tr>
<td></td>
<td>Resource pooling</td>
<td>Tang et al. (2021), Sorkhoh et al. (2020), Cha et al. (2021) and Hameed et al. (2021)</td>
<td>Scalability/reliability/eval. w. rwmds.a</td>
</tr>
<tr>
<td>VFC 51%</td>
<td>Task replication</td>
<td>Sun et al. (2021)</td>
<td>Reliability/collaboration/eval. w. rwmds.a</td>
</tr>
<tr>
<td></td>
<td>Multi-period task offloading</td>
<td>Liu et al. (2021)</td>
<td>Reliability/eval. w. rwmds.a</td>
</tr>
<tr>
<td></td>
<td>Offl. under inform. incompletn. and uncertainty</td>
<td>Zhou et al. (2019)</td>
<td>Context-awareness/eval. w. rwmds.a</td>
</tr>
<tr>
<td></td>
<td>Semantic-aware real-time data processing</td>
<td>Fu et al. (2021)</td>
<td>Scalability/redundancy</td>
</tr>
<tr>
<td></td>
<td>Event-triggered offloading and migration</td>
<td>Zhu et al. (2019b)</td>
<td>Scalability</td>
</tr>
<tr>
<td></td>
<td>Improv. net. connectivity w. forward. and proxy.</td>
<td>Liang et al. (2021), and Jabri et al. (2019)</td>
<td>Scalability/reliability/security</td>
</tr>
<tr>
<td></td>
<td>Overloaded node management</td>
<td>Yadav et al. (2020)</td>
<td>Scalability/reliability/collaboration/eval. w. rwmds.a</td>
</tr>
<tr>
<td></td>
<td>Microservices</td>
<td>Sami et al. (2020)</td>
<td>Node heterogeneity/recovery performance/security</td>
</tr>
<tr>
<td></td>
<td>Node redund. to improve computing reliability</td>
<td>Dong et al. (2020)</td>
<td>Scalability/eval. w. rwmds.a</td>
</tr>
<tr>
<td></td>
<td>Modelling the effects of mobility</td>
<td>Saad and Grande (2020)</td>
<td>Scalability/eval. w. rwmds.a, pne.b</td>
</tr>
</tbody>
</table>

aReal-world mobility data sets.  
bPractical networking environments.

expectations at most (Table 5). Remarkably, that has roots in the purely theoretical nature of those works, which typically leaves aside tests in practical networking environments. Going even further, insufficient testing efforts lead to mediocre exploitation of multi-tier cooperation.
and collaboration, which in turn are expected to improve the reliability and continuity of the offered fog services. On the other hand, if we look at the percentage coverage of the topics within the models (Fig. 4), it becomes even more evident that certain aspects are deliberately left as afterthoughts by researchers. Sadly, only a few solutions which claimed to be fog instances were devised with awareness of OpenFog concept. Another observation emphasises a great variety of solution targets undertaken by the authors of the reviewed papers. This has an effect on the low coverage of some of the topics (Table 5). Resource pooling and cooperation during offloading have decent coverage (at least in VFC), while functional safety aspects like redundancy and replication still require work. From the methodology point of view, one-third of all the papers rely on heuristics (Fig. 5, Fig. 5(a)), and even more of them set latency minimisation as the optimisation goal, Fig. 5(b). While reduction of delay seems to be the natural choice for fog computing, a huge share of algorithms with fragile mathematical foundations that require sophisticated tuning do not bode well for deployments.

6.7. Further research

Solutions for fog computing should undoubtedly explore the mobility of fog nodes in providing robust services to its users. It is expected that upcoming works either focus on careful construction of the reliable FC components, or combine the achievements from various FC models to devise scalable systems with a slight number of implicit features. In particular, following aspects should attract researchers’ interest:

- **Robustness.** New solutions should tolerate range of discrepancies without negative impact on the behaviour and observed performance (in both QoS and QoE). Researchers are encouraged to clearly indicate the parties obliged to run the devised algorithms, then provide both the computational complexity analysis and thorough performance evaluation. Performance should be measured in the long run using highly heterogeneous setups (at the node and the network levels) and diversity of behaviours (e.g. spontaneous decisions, sudden disruptions, density peaks, etc.). Simulations of mobility should be based on real mobility data sets, which are either publicly available or shared together with the respective article, to enable cross testing and comparisons.

- **Interoperability.** Implementations of fog computing are expected to coexist with extant systems and rely on the firmly established features. Hence, more attention should be paid to ensure and verify the secure information exchange. Example scenarios include integration of third-party solutions (e.g. trust anchors, semantic and social data providers, etc.), benefiting from device and communication media heterogeneity, distribution of decisive power among nodes and the hierarchy, and inclusion of the existing networking and computing systems.

- **Composability.** Decomposition of complex ideas into interacting entities of high cohesion should be applied to the forthcoming architectures for FC-compliant systems. Therefore, more work is expected in design of mature pluggable components that adapt their behaviours to context changes (i.e. have low decision inertia and reduce possibility to be driven by stale data). Collaboration, rather than cooperation, between the opaque entities over the newly standardised interfaces should become a main concern in providing scalable designs and healthy architectures.

7. Conclusion

This survey discusses mobility-aware fog computing with mobile fog nodes in highly dynamic environments, where the time-varying modifications of the fog network topology are present. Five mobility-aware fog computing models of the fog computing concept compliant with OpenFog adoption as in the IEEE 1934 standard document are identified and described. The conducted systematic review study adopts the multi-aspect broadened perspective on fog computing to highlight the coverage of the OpenFog pillars that are crucial for fog computing with mobile nodes. As discovered, the reviewed papers address the arbitrary aspects of the problem with varying quality and consistency. Therefore, further research is urgently required to compose the complete solutions out of the building blocks designed by various authors, rather than starting from ground zero. Authors believe that, as with mobile fog computing and mobile nodes, agile collaboration is a key here.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

Abboud, Khadige, Zhuang, Weihua, 2014. Stochastic analysis of a single-hop communica-