Mobile computing and communications-driven fog-assisted disaster evacuation techniques for context-aware guidance support: A survey

Kurniawan, I. F., Asyhari, A. T., He, F. & Liu, Y.

Original citation & hyperlink:
Kurniawan, IF, Asyhari, AT, He, F & Liu, Y 2021, 'Mobile computing and communications-driven fog-assisted disaster evacuation techniques for context-aware guidance support: A survey', *Computer Communications*, vol. 179, pp. 195-216.
https://dx.doi.org/10.1016/j.comcom.2021.07.020

DOI 10.1016/j.comcom.2021.07.020
ISSN 0140-3664

Publisher: Elsevier

Mobile Computing and Communications-Driven Fog-Assisted Disaster Evacuation Techniques for Context-Aware Guidance Support: A Survey

Ibnu Febry Kurniawan[a,d,1,*], A. Taufiq Asyhari[b], Fei He[a,c], Ye Liu[a,c]

[a]*Centre for Data Science, Coventry University, Coventry, CV1 2JH, UK.*
[b]*School of Computing and Digital Technology, Birmingham City University, Birmingham, B4 7XG, UK.*
[c]*School of Computing, Electronics and Mathematics, Coventry University, Coventry, CV1 5FB, UK.*
[d]*Department of Informatics, Universitas Negeri Surabaya, Surabaya, 60231, Indonesia.*

Abstract

The importance of an optimal solution for disaster evacuation has recently raised attention from researchers across multiple disciplines. This is not only a serious, but also a challenging task due to the complexities of the evacuees' behaviors, route planning, and demanding coordination services. Although existing studies have addressed these challenges to some extent, mass evacuation in natural disasters tends to be difficult to predict and manage due to the limitation of the underlying models to capture realistic situations. It is therefore desirable to have on-demand mechanisms of locally-driven computing and data exchange services in order to enable near real-time capture of the disaster area during the evacuation. For this purpose, this paper comprehensively surveys recent advances in information and communication technology-enabled disaster evacuations, with the focus on fog computation and communication services to support a massive evacuation process. A numerous variety of tools and techniques are encapsulated within a coordinated on-demand strategy of an evacuation platform, which is aimed to provide a

*Corresponding author

Email addresses: `ibnufebry@unesa.ac.id` (Ibnu Febry Kurniawan), `taufiq-a@ieee.org` (A. Taufiq Asyhari), `fei.he@coventry.ac.uk` (Fei He), `ye.liu@coventry.ac.uk` (Ye Liu)

[1]The work of I.F. Kurniawan was supported by the Directorate General of Resources for Science, Technology and Higher Education, Indonesia.

situational awareness and response. Herein fog services appear to be one of the viable options for responsive mass evacuation because they enable low latency data processing and dissemination. They can additionally provide data analytics support for autonomous learning for both the short-term guidance supports and long-term usages. This work extends the existing data-oriented framework by outlining comprehensive functionalities and providing seamless integration. We review the principles, challenges, and future direction of the state-of-the-art strategies proposed to sit within each functionality. Taken together, this survey highlights the importance of adaptive coordination and reconfiguration within the fog services to facilitate responsive mass evacuations as well as open up new research challenges associated with analytics-embedding network and computation, which is critical for any disaster recovery activities.

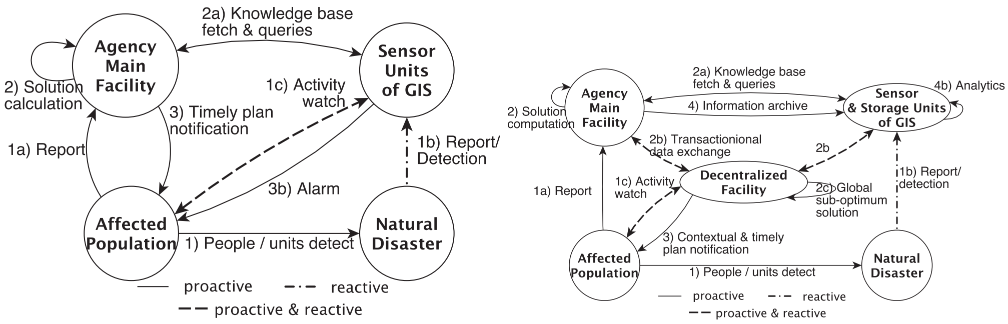
Keywords: Disaster recovery, evacuation guidance, fog, fog computing, fog communications, collaborative analytics

---

1. Introduction

Large-scale evacuation solutions for pre-, on-going, and post-disaster natural disaster events have been studied extensively in recent years. Devising an effective evacuation solution is a critical and also a complicated task because it relates to the number of casualties and mobilization management. These solutions should consider several important factors, such as road capacity, human behavior, capability [1, 2, 3], and infrastructure destruction [4]. Moreover, these developed solutions often uniquely serve for different phases of a disaster. Pre- and on-going disaster evacuation strategies govern massive mobilization to reduce travel time, and plan the best route, while post-disaster evacuation aims at locating survivors quickly within the golden relief period. Due to its specific condition of each disaster part, and related factor complexities, a large body of knowledge has utilized various approaches that can be mainly categorized into agent-based modeling (ABM) [1, 3, 5, 2], and crowdsourced services [6, 7, 8, 4, 9].

Despite many benefits given by the ABM strategies, they lack the ability to adequately react to unexpected situation due to the model used. The optimal solution of these strategies are pre-computed (Fig. 1a) and tend to be less-realistic due to various dynamic elements, such as weather change, unexpected traffic surge, road network destruction, and evacuation rate [4]. Also, the usage of the agent-based model tends to be location-, and population-specific due to the assumptions made in the system modeling step. Motivated by the static nature of the agent-based model, crowdsourced service has been designed to provide an on-demand solution using a robust infrastructure. The service processes data input at the designated cloud facility, and outputs the solution to information outlets, such as roadside units [4], or handheld devices [6, 8, 10]. However, these solutions are highly likely to suffer from transmission delays caused by the data propagation, and queuing process. Moreover, the lack of a failover plan caused by physical damage or system failure at the core layer of the facility makes this approach alone more vulnerable to failure [7, 10].



(a) with Agent-based model OR GIS

(b) with Fog service

Figure 1: Network-enabled computation framework for disaster evacuation.

Fog computing and communication is a technological approach that has the potential to address high-delay issues in centralized service [11] by bringing the computation facility closer to end-users. Existing studies [11, 12] showed several substantial contributions to low-latency, localized solutions, and low-cost deployment. Furthermore, these studies combined the remaining functioning infrastructure and crowd-based services, producing a viable solution for crowd management in a disruptive scenario. Not only does it offer a more fault-tolerance strategy, but also provide a circumstantial awareness that allows authorities to observe various crowd conditions [8]. This service can typically consist of several mobile computing units (MCUs), such as movable fog nodes, smartphones, and sensing devices [11, 12], as well as mobile assistant units (MAUs), namely Unmanned Aerial Vehicles (UAV) [9, 7, 13] and Unmanned Ground Vehicles (UGV) [14]. These sub-units can communicate and work together for data collection [9, 14] or short-term decision-making [15]. An autonomous collaboration between the ground and aerial units can improve the quality of imagery data. As a result, fog-computing-aided crowdsourced service is envisioned to produce optimum evaluation paths by considering the density of the road and traffic congestion. The platform has sufficient capability to interpret and process real-time situations locally and forwards brief key information to the central units, if possible. The computation process can be further improved by taking into account historical and geographical information surrounding the catchment areas [16, 17, 18, 19, 20]. A generic illustration of this approach can be found in Fig. 1b.

Table 1: Comparison of related surveys.

Source	Main highlight
Mukherjee *et al.* [21]	Fundamental fog architecture, key metrics, challenges, and applications.
Mahmud *et al.*[22]	Key differentiating concepts between other new computing paradigms, and fog taxonomy based identified challenges.
Mouradian *et al.* [23]	Comprehensive survey on fog computing based on architectural and algorithmic perspective.
Hu *et al.* [24]	Survey on key technologies and applications of fog computing.
Our work	Novel framework-based survey on fog computing key-enabling technologies aiming at seamless integration specifically for evacuation service.

In addition to the abovementioned benefits, the use of mobile fog services also highlights the potential of data analytics implementation in which the existing solution lacks. The collected data and the stored knowledge base, such as sensing data and trajectory patterns will provide valuable direction on how the solution should be devised and implemented. Additionally, the capability to analyze and derive network condition pattern, such as transmission loss [7] and channel capacity [10], offers the potential to do reactive reconfiguration. However, finding a balance between these metrics is a challenging issue because of the limited energy which mobile fog and terminal devices have [25, 26, 27]. Mobile fog is different from a typical fog device because it only relies on battery to serve any incoming requests [11]. Therefore, not all types of data should be considered important and served equally [7]. Beside the energy issue for the computation purpose, collaborative processes that occur between all units should be carefully governed. It causes not only communication interference but also a queuing problem.

Though there have existed several surveys of fog computing as listed in Table 1, this paper serves different purposes. Previous surveys [21, 22, 23, 24] cover fog computing concepts with provided taxonomy. In contrast, this paper surveys the state-of-the-art fog-assisted technologies with scalable communications and data analytics for context-aware disaster evacuation scenarios. Our contributions are summarized as follows:

- *Top-down comprehensive review.* Our survey is structured based on an umbrella of a fog-based service architecture named Fog-Assisted Disaster Evacuation Service (FADE). We illustrate FADE as a conceptual stack of logical capabilities that mainly operate at the mobile computing unit.

- *Critical evaluation on communication technology.* We formulate an integrated communication service suitable for high-demand access during an emergency situation. Then, we analyze the building elements for a low-delay communication context.

- *In-depth examination on analytics capability.* We list and review the practical applications and examine possible technological adoptions to anticipate uncertain conditions.

- *Discussion on technology interoperability.* We extend the original layered-oriented Data-driven Architecture [28] by incorporating comprehensive functionalities on each layer and providing a seamless integration.

- *Highlight on critical challenges and future directions.* We discuss current research challenges for each supporting technology and provide brief directions to address them.

The remaining parts of this paper are organized as follows. Firstly, we review the existing evacuation solutions in Section 2. Then, the key features and architecture of fog-assisted service, including high-level perspective and layered-oriented functionalities, are presented in Section 3. In Sections 4-7, existing works on each functionality component are discussed. Since FADE offers a collaborative computation feature, a functionality for retrieval and analytics requires modification of the corresponding layers. These are discussed in Section 8. Then, we present the research challenges and future direction of related topics in Section 9. Finally, we conclude our work in Section 10 with an overview of our contributions.

2. Existing Evacuation Guidance Solution

This section compares several existing evacuation guidance approaches that focus on different phases of a disaster event. This highlights several important aspects that differentiate the principles used in the existing strategies with the envisioned collaborative method. In brief, the main distinctive aspect between the strategies covered in Table 1 and FADE is the presence of a decentralized facility to enable a lower-latency coordination and a future analytics.

Table 2 presents available strategies for evacuation guidance or route planning that can be classified into different categories. Herein a strategy for delivery indicates an appropriate phase where solution calculation or dissemination occurs. Pre-computed strategies attempt to compute an optimal evacuation plan before the disaster happens. Meanwhile, the strategies with on-demand approach (Fig. 1b) compute the optimal evacuation path requested during the occurrence of natural disaster. These stragies of delivery can utilize one of two distinct information flows either uni-directional or bi-directional. While uni-directional strategies use pre-stored data and deliver to authorities, bi-directional schemes require field data captured multiple times by on-field units before finally sending the optimal decision to emergency response units. The next category, focus of disaster event, indicates which part of a disaster event aimed by an evacuation guidance scheme. The subsequent category is contribution areas which lists the focus of each strategy, such as modelling, simulation, infrastructure, and communication aspect.

The ABM strategies provide an approach to simulate and to evaluate the decision taken by a trained personnel to mobilize the affected population before and during the occurrence of a disaster. In this approach, the agents often have better knowledge than residents such that they possess capabilities to decide the best route to safe points. The authors of [1] and [3] devised an agent-based strategy that considers pedestrian's capability, road capacity, and moving direction. Similarly, the authors of [5] and [2] designed an algorithm that manages evacuation from a macro and micro-level perspective. The second approach, crowdsourced service, extends or replaces the existing communication infrastructure with users' devices, and sensor networks. This service exploits the short-range information exchange between the users' equipment to provide a backup communication service using Delay-tolerant Network [6] or Opportunistic Communication [7]. Additionally, the crowdsourced service also can offer additional data capture source [8, 4], and cognitive resource [11, 9] to help the complex computation task.

As seen in Table 2, most of the existing evacuation guidance approaches and plans were developed using pure [2, 3, 29, 30, 1] or hybrid agent-based [5, 31] models. In these pure agent-based solutions, a simulation tool was

Table 2: Existing Solutions to Disaster Evacuation Problem

Ref.	Delivery strategy	Info. flow	Focus of disaster event			Contribution area(s)	Key concept(s)
			Pre-	During	Post-		
[1]	P	1	✓	✓	✗	*M*,*S*	ABM†
[3]	P	1	✓	✓	✗	*M*,*S*	ABM†
[5]	P	1	✓	✓	✗	*M*,*S*	ABM‡
[2]	P	1	✓	✓	✗	*M*,*S*	ABM†
[29]	P	1	✓	✓	✗	*M*,*S*,*I*	ABM†
[30]	P	1	✗	✓	✓	*M*,*S*	ABM†
[31]	O	2	✗	✗	✓	*M*,*I*	ABM‡, DTN
[10]	O	1	✓	✓	✗	*M*,*S*	Routing
[4]	O	2	✓	✓	✗	*M*,*I*	Cloud
[12]	O	2	✓	✓	✓	*I*	Fog
[11]	O	2	✓	✓	✓	*I*	Fog
[8]	O	2	✓	✓	✓	*C*,*I*	Cloud, Fog
[6]	O	1	✗	✗	✓	*C*,*I*	DTN
[32]	O	2	✗	✓	✓	*M*,*S*,*C*	DTN
[13]	O	2	✓	✓	✗	*M*,*I*	UAV
[9]	O	2	✓	✓	✗	*M*,*I*	UAV
[7]	O	2	✗	✗	✓	*M*,*S*	Cloud, UAV
[33]	O	2	✓	✓	✓	*S*,*I*	UAV
[34]	O	1	✗	✗	✓	*M*,*S*	UAV
[35]	O	2	✗	✓	✓	*S*,*I*	WSN
[36]	O	1	✗	✗	✓	*S*,*C*	WSN
[37]	O	2	✓	✓	✓	*I*	Cloud, SDN

Note. *P* = pre-computed; *O* = on-demand; *1/2* = uni-/bi-directional; *M* = modelling; *S* = simulation; *I* = infrastructure; *C* = communication; †/‡ = pure/hybrid.

used to investigate the impact of the agents' parameters, such as horizontal, vertical evacuation, multimodal capability, and evacuees' decision time, on the total number casualties. Some important findings were obtained from these studies, such as: (1) mortality rate was sensitive to the time delay; (2) variation in walking speed caused by multi-modal evacuation determined the number of casualties due to the traffic congestion; and (3) shelter allocation for vertical evacuation could greatly reduce the mortality rate. Due to these assumptions made on the model, the running time of the simulation increases. Motivated by these limitations, a parallel algorithm was proposed in [29] to speed up the computation and executed in High-performance Computing (HPC) infrastructure. This work simulated agents for evacuation with visibility parameters to recognize the busy road situation and react accordingly to the condition.

While the aforementioned methods demonstrated a potential reduction in the disaster mitigation time, they were static, making them less responsive to unexpected situations. In fact, mitigation time is a complex metric as it is related to the completion time of various disaster relief processes, such as casualty loss estimation, evacuee behavior analysis, traffic bottleneck identification, shelter condition monitoring, and evacuation mode evaluation [3]. Because the pre-computed strategies lacked of pro-active and reactive response mechanism to adapt with various scenarios, a more dynamic approach was then proposed in [4, 9, 13]. In these studies, a two-way data exchange was used for an input and output mechanism that facilitated a series of interactive evacuation guidance. MACROSERV [4] incorporates a joint function between Intelligent Traffic System (ITS) and sensing system. The platform acts as: (1) an evacuation-plan analysis—tool used by authorities that simulates various disaster scenarios–; and (2) an efficient-route recommendation tool during the course of a disaster. The service processes real-time traffic data provided by Road-side Units (RSUs), then transfers them to a centralized computation facility, and outputs the generated optimum route to the RSUs. The MACROSERV optimizes an evacuation route by considering the models of the road capacity, traffic volume, route distance, and population size. Meanwhile, in order to provide similar responsiveness in obtaining input and delivering instruction, several works, i.e., [6, 19, 11] used aerial units to replace human agents. These UAV units have more mobility, making them capable of gathering evacuation data [6] to guide the evacuees according to a pre-computed plan [13]. In [13], the drones were used as a replacement of human-agents to guide evacuees to safe location points.

In addition to a desirable dynamic support provided by the evacuation guidance, computation capability is also seen as an important aspect to focus on. Not only does it affect how well the response is computed, but also how effective it is to operate during the evacuation process. This consideration becomes more relevant when a multi-hazard disaster scenario occurs, such as the 2018 Sulawesi [38] and the 2010 Chile [39] earthquake-tsunami. For this reason, several studies designed a hybrid platform including infrastructure and crowdsourced facilities to alleviate the computation demand [11], to backup the barebone network service [8, 6], or to provide the emergency communication service [7]. The centralized computation facility that offers a powerful computation capability was used in [4] to help the decision-making process. However, the support comes with the cost of a longer response time and thus makes it less desirable in a highly demanding and low-latency scenario [11, 12]. Therefore, the notion of *Opportunistic Fog* was proposed in [11] as a movable computation facility aiming at localized solutions and low-latency data transfers. Meanwhile, an integrated communication acting as a scalable service during a crowd was proposed in [8, 6]. In these studies, the authors designed a communication network that enabled a combination of peer-to-peer connection of user devices and infrastructure-based service. The user-to-user connection establishes an opportunistic concept that is prone to high-latency and low-delivery rate.

3. Fog-assisted Disaster Evacuation (FADE) Solution

3.1. Principle Characteristics

While conventional strategies have predominant characteristics of being uni-directional, static, and location-specific solutions, FADE is envisioned to create a bi-directional flow of information that is intended to cope with the dynamic nature of the evacuation process. Additionally, the existing solutions tend to be pre-computed and use non-stationary computing infrastructures. Key technologies to enable FADE functionalities are mobile fog infrastructure and aerial surveillance units, which were proposed by a number of pilot studies [11, 9, 12]. According to [11], opportunistic fog provides high-quality services to a wider area with higher localized accuracy compared to the cloud processing via low-latency and transient connectivity. We describe the unique features of FADE compared to the conventional solutions as follows.

- *On-field data capture*: FADE attempts to monitor on-going evacuation activities via ad-hoc infrastructures under the guidance of the central processing unit. These on-demand supports are expected to arrive on the affected population ideally before the disaster fatalities emerge. The first impact time of several disasters such as tsunami, wildfire, hurricane, or flood can be predicted and spans over 20 minutes to 1 hour prior to the event [1]. During this critical time, a careful traffic management considering several key factors, such as road capacity, movement direction, and safe points, is necessary. Different methods and data can be used to obtain this information, such as aerial images [13], sensor data [4], or phone signal activities [31, 7].

- *Online solution*: A conventional strategy on the evacuation guidance mainly considers the usage of a pre-computed contingency plan that will be executed by the evacuees [1, 3] or the rescue team [2, 13]. In this case, the solution is calculated as an output of a problem formulation under which the model may have some limitations [4]. Firstly, the conventional algorithm uses a static road model that is unable to capture the real emergency scenario. Secondly, the assumptions used in the model can be less relevant due to variations in weather, unexpected conditions of traffic, and physical damages to the infrastructure. Fig. 1a illustrates how the offline solution's framework operates. Sensing inputs from a Geographic Information System (GIS) sub-units act as an early detection system and a regular data capture. On the contrary, FADE provides live assistance computed by Mobile Fog Units (MFU) and subsequently disseminate processed information via MAU instructions.

- *Predictive step*: The affected population has unique and recurrent mobility patterns that can be possibly obtained from timestamped GPS [17, 40] or mobile phone cell records [41]. This information combined with the GIS data stores can provide sufficient training data to predict the mobility pattern of local residents. Upon receiving the alarm for the detection system, FADE will identify the central position of a disaster event and attempt to map which part of the affected location should be prioritized. This prioritization necessitates the computation of the residents' trajectory data and, thus, can be used to avoid traffic congestion. Furthermore, during the evacuation process, field data will contribute to the prediction as additional training data, which serve as valuable inputs to the routing optimization process. Also, these additional data can be used for training or stored in the central unit for future usage. See, e.g., Fig. 1b for illustration.

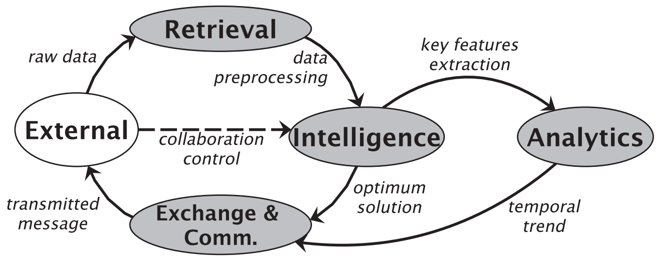


Figure 2: A generic cognitive process diagram for FADE.

3.2. Management of Functionalities

FADE performs complex computation using various pre- and during disaster data inputs provided by mobile units, sensing devices, and local authorities. These data are then propagated to local gateways provided by the MFU. Due to a variety in circulated data types, traffic management is needed to avoid the bottleneck in the system. Depending on the computation scheme used in the computation facilities, these data are then distributed within the computing clusters, processed, and finally stored at the end of the evacuation process. Due to the high frequency of data exchange, processing and analytics, FADE adopts Data-driven Architecture proposed in [28] for context-aware guidance support. The framework governs the data cycle starting from retrieval to mining activities, which are an ideal base for a scalable data management.

FADE proposes management of data-driven functionalities by creating an overlay connection to four activities within the initial Data-driven architecture. This setup extends the original capability of the architecture to suit FADE core processes. Each functionality represents a unique processing block and will be mapped to a related layer(s) in this architecture. Fig. 2 depicts a high-level information cycle involving FADE internal sub-systems and external actors. Based on this cycle, the connection and information exchanged between each process is drawn in Fig. 3. There are four proposed functionalities encapsulated into functional layers, which are described further as follows.

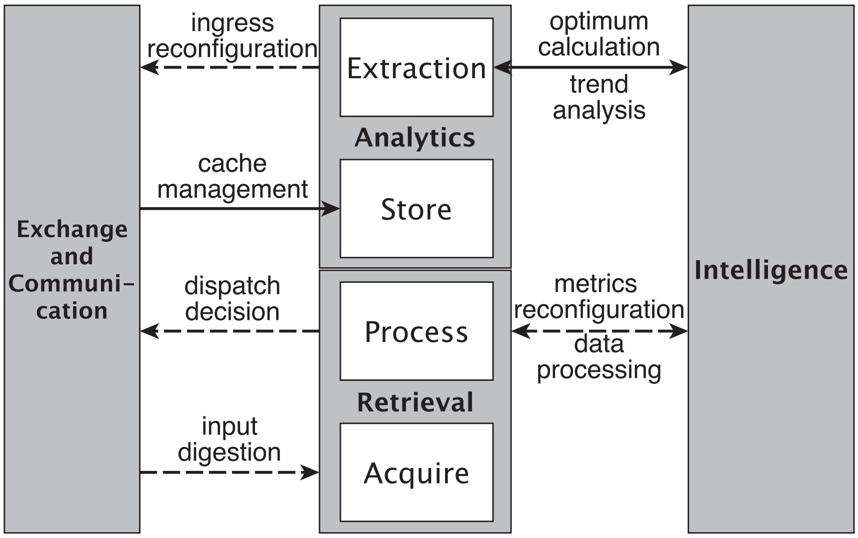


Figure 3: The four functionalities (functional layers) and their interactions in FADE.

- *Retrieval*: This functionality handles the early stage of the process where multiple data types from various data sources can contribute. There are two sub-processes herein, namely data collection and preparation. The first sub-process governs how data are retrieved by mobile units and computing units, whilst, the second sub-process pre-processes the inputs for subsequent steps. This later sub-process is required because different data types can proceed through different activities. Therefore, the processed data should match the requirement of the next process.
- *Exchange and Communications*: To support collaborative processing characteristics of FADE, incoming and outgoing data managements are important. This functionality governs data distribution in the field capture, mobile unit communications, or long-term data store activities. Due to the high amount of transmitted data, this functional layer should be able to handle scalable data growth, which can be achieved by a configuration that is provided by the Intelligence functional layer.
- *Intelligence*: Parameter-based computation using physical and theoretical models takes place in this functionality. Live data obtained during the evacuation process, e.g., pedestrian images, latest environmental

data, or network usage, are processed using various techniques, e.g., crowd analysis or network traffic estimation, to obtain relevant information and optimize internal system configuration. This functionality requires robust input processing, image or text data processing, and a smart mechanism to manage traffic allocation to avoid congestion.

- *Analytics*: Data-driven techniques for long-term purposes, such as data training and learning, take place in this functionality. The former operation focuses on forecasting possible user behaviors based on the historical data. Meanwhile, the latter mainly concentrate on storing and analyzing data for future trajectory modeling, user behavior recognition, or network metrics profiling. This functionality is motivated by the fact that computation can be done by separate units of multi-tier systems, such as fog or cloud architecture, which can have more resources.

Security and privacy are important aspect but not the main concern of FADE. These two features can affect both the system's performance and availability at normal and emergency time. During non-disaster time, FADE collects environmental telemetry and post-evacuation log analysis, which can be retrieved from third-party data providers, such as meteorological office, intelligent transport system, or government agency office. Such a data exchange takes place on the backbone network which is protected by enterprise-level security protection. On the other hand, privacy and security processing during disaster evacuation should be kept to minimum as it potentially deteriorates time-sensitive communication and energy-constrained computation of mobile fog infrastructure.

4. Retrieval Functionality in FADE

4.1. Data Ingestion

Data ingestion activity manages how data is received from an external data storage or a raw data capture. As illustrated in Fig. 1b, FADE can receive data from two sources, namely existing GIS databases and deployed MAUs. While the first source provides on-premise computing facilities with the stored historical data, the second source pushes on-field data via ad-hoc wireless communication networks. We refer to the corresponding first and second data categories as the stored data and live data, respectively.

Table 3: Summary of FADE functionality key action and reference.

Main and sub-functionality	Activity Overview	Key Reference(s)
A. Retrieval		
A1. Data Ingestion	Primary and secondary-source data input administration.	[42, 43, 44, 45, 46, 47, 9]
A2. Data Preparation		[48, 49, 50, 51, 52, 53, 54, 55]
B. Exchange and Communication		
B1. Information Broker	Internal and external transactional data circulation management.	[56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66]
B2. Cache Management		[67, 68, 69, 70, 71, 72, 73, 74]
C. Intelligence		
C1. Metrics Processing	On-demand optimization process for evacuation guidance and internal system setup.	[75, 76, 77, 78, 79, 80, 81, 82, 83, 79, 84, 85]
C2. Optimum Calculation		[86, 87, 88, 89, 90, 91, 92, 93, 94, 1, 95]
D. Analytics		
D1. Reconfiguration	Data-driven learning process for long-term and case-by-case configuration.	[96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107]
D2. Trend Analysis		[16, 17, 18, 19, 20, 40, 108]

- *Stored data*: Two types of data can help evacuation guidance for FADE, i.e., the IoT-provided sensor data [42, 43, 44] and the trajectory data model [17, 41]. FADE can gain valuable information on geographical conditions and people's mobility. Sensor data will provide FADE historical metric data of the affected locations, including water level, wind speed, or sound. These readings can help FADE understand which part of the affected location that is heavily or will be likely affected by a natural disaster. Meanwhile, the trajectory data contain the movement pattern of local residents during a certain period and are made available as an output of the movement analysis [17, 41]. FADE will retrieve the training model for some parts of the evacuation point.

- *Live data*: The live data category consists of the raw data of the evacuees and environment, including the road and shelter conditions during the evacuation process. Different data types can be utilized for this purpose, including aerial images [11, 47], cellphone signals or activities [7, 109], and even video inputs [46, 45]. These data will be further processed by the Intelligence layer to derive the road density, to identify the impassable roads, and to manage the distribution of the evacuees to each shelter.

The mobile unit can collaboratively obtain an evacuation image and upload it to its controlling device shortly. This is made available via an established socket connection between mobile and control units. Then, the controlling unit can distribute more complex processing activities to portable computing units as shown in [9]. Cellphone activities can be made available from radio observations within the evacuation scene. A radio unit, such as a cellphone or radio modem, can be attached to a mobile agent and capture the signal activities based on the transceiver activities or feature detection [110].

Realizing potential complexities rises from these two data types, both the data receving and operation process must be managed intesively. A coordination between these two sub-functionalities and other higher functionalities namely Exchange and Intelligence is required (Fig. 4).

4.2. Data Preparation

Data received by each relevant unit have different types and require different preparation strategies. Data collected by the on-field device units can be blurry, redundant, or lacking of pedestrian images due to bad angle captures, weather conditions, or inaccurate deployment positions. On the other hand, the data series provided by government database can be incomplete or erroneous because of unreliable transmission. Data preparation in these contexts can be generally classified into three main groups, namely data cleansing, integration, and reduction as investigated in [111]. In FADE, cleansing is envisaged to deal with incompleteness and erroneous conditions. Integration manages data enhancement whereas reduction controls feature and input reduction. We then further map the strategy to a possible execution in the fog environment as seen in Table 4.

One of core FADE's activities that rely on good-quality data is crowd counting [112]. This cognitive mechanism, will be further discussed in Section 6.2.1, analyzes the evacuation process by counting the number of pedestrians and estimating the road density from a single image input. Then, the input will be computed by learning methods, such as Deep Learning. In such a technique, image data will be computed via several layers of computation. The deeper the data proceed, the more complex the process. If the input has too many noise or missing parts, then the results' accuracy and precision will

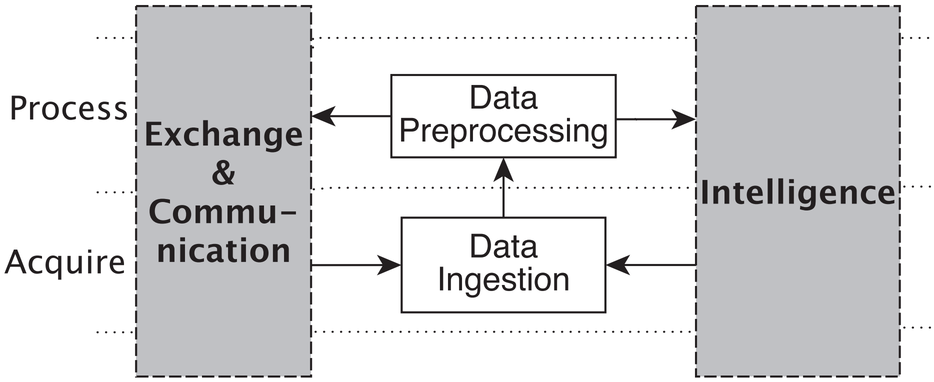


Figure 4: Detailed processing within the Retrieval functionality.

be lower. In other words, Deep Learning-based technique is highly reluctant to the quality of dataset. The work in [112] reviewed a wide range of strategies that only run on a single computing unit and can be further improved by distributed computing as discussed in [113].

In view of the distributed processing strategy, the implementation of collaborative data preparation depends on the execution mechanism or the topology setup of FADE. Such a process can be done by an equal or hierarchical distribution that may focus on the item or step execution. Table 4 lists three different implementations, namely non-, peer- and hierarchical-offloading. A non-offloading or single device computation scheme performs all preparation tasks by each mobile units and is particularly suitable for error and incomplete data detection and lightweight image preprocessing. Meanwhile, peer-offloading arranges the end-devices to prepare inputs together so that each device has a level of involvement. The third strategy, namely hierarchical-offloading, regulates the data preparation based on the level of difficulties where lower and higher layers will be assigned to end-devices and servers, respectively. This setup is designed by dividing each process complexity in which a higher layer has more matrix multiplication and requires higher computation resources.

4.2.1. Non-offloading data preparation

Data preparation conducted by the on-field unit should be lightweight to minimize the computation time because of the limited energy resources. Several simple image processing activities, i.e., grayscale conversion, image enhance-

Table 4: Data Preparation Strategy

Strategy	Image		Text	
	Operation	Execution	Operation	Execution
Cleansing	Abnormality detection [53]	○, ☽	Abnormality detection [53]	○, ☽
	Background removal [49, 53]	○		
	Filter application [49]			
	Enhancement [48]			
Integration	Model partition [114, 115, 116]	●	Model partition [114, 115, 116]	●
			Discretization [54]	☽
Reduction	Feature extraction [54]	☽	Feature reduction [54, 55]	☽,●
	Object localization [117, 50]	○	Dimensionality reduction [55]	●
	Compression [49, 53]	○	Instance reduction [114, 115, 116]	●
	Grayscale conversion [48]	○		

Note. ○ non-offloading; ☽ peer-offloading; ● hierarchical-offloading

ment [48], or filter application [49], are suitable strategies for this purpose. More complex image operations such as the early stage of image classification [50], feature extraction [117], or even inference ability [118, 51, 52] can be run on mobile devices with more power and energy resource. Additionally, fast detection of the data quality is also important as it will affect the final result. Adversarial or corrupted data should not be further processed and on-field data re-capture should be executed as soon as possible. Strategies such as aberrant data detection [53] are suitable for this purpose because of its low complexity.

4.2.2. Peer-offloading data preparation

This strategy often has higher complexity because it turns local data from its raw form into communal information. A series of data will be initially captured by each mobile unit. The data will then be shared before combined or summarized. The types of textual data collected are not necessarily identical because this may also include data integration [111]. However, preprocessing applied to the collected data requires computation resources if performed individually and locally. Therefore, a participative strategy or task sharing across mobile units is needed to mitigate the cost as discussed in [113]. One of applicable activity in participative strategy is data discretization [54]. It is one of the feature space simplification techniques [55] where continuous data are summarized into a fixed set of intervals. This strategy aims to associate numerical values with certain intervals and is commonly found in supervised learning. Pedestrian activities and environmental conditions during an evacuation can be categorized using supervised learning. The attribute and data samples of the evacuation process can be determined and aggregated, respectively.

4.2.3. Hierarchical-offloading data preparation

The hierarchical strategy regulates how data are prepared by performing task assignments. In this activity, a task will be decomposed into less-complicated sub-tasks and will be then delegated to different computing units. Compared to previous offloading strategies, this approach will no longer output the enhanced data, but rather learning information. In the context of Deep Learning, the learning process consists of interconnected layers performing different computations. The output of one layer becomes the input of the next layer. As an example, consider the application of Convolutional Neural Network (CNN) on crowd counting activity. CNN is one of machine learning techniques that uses the properties of natural signal, such as shared weights, pooling, and layer stacking [92]. The strategy comprises of layers of convolutional operation blocks that operates on different portion of input data. According to [112], the more advanced an approach is, the more layer the approach has. In such a case, due to the complex mechanism and the depth of processing layers, relying on the central unit will result in a long processing queue leading to high latency.

In order to reduce the running time of learning processes, some studies [114, 115, 116] designed strategies to decompose and allocate the workload to end-devices and central servers. The workload allocation can be achieved either by implementing static [114] or flexible partitioning [115, 116]. While the former strategy distributes learning tasks to end-devices and computing facilities based on its complexity, the latter one reconstructs learning layers so that end-devices can work in parallel. Although both approaches are suitable for low resource devices, the second one leaves a smaller memory footprint and less communication cost. FADE can adopt this mechanism so that UAVs can compute the data more independently. This is also relevant where communication channels during a natural disaster can be less reliable or even not available.

5. Exchange and Communication Functionality in FADE

5.1. Information Broker

Disaster management is an exhausting process where diverse and expensive computing tasks are performed by various fog service components. Interaction between elements, shown in Fig. 5, is crucial and requires scalable data exchange management. In this section, we refer Information Broker to a sub-functionality that controls how data are circulated during evacuation. Communications that occur during the process involves the acquisition and process layers in the Data Management framework. We explain the various communication (data exchange) modes in FADE as follows:

- *Mobile-to-mobile (M2M) units communications*: This communication scheme covers two types of data transmission, namely coordination and transactional messages. While the former packets comprise of navigation controls, relayed command messages, or evacuation instructions, the latter includes raw and pre-processed data as explained in Section 4.2. The communication that takes place in this mode tends to be intensive and vital to the mission's success. On-field data are important to analyze the effect of disaster and to predict evacuees' mobility pattern.

- *Mobile-to-processing (M2P) units communications*: A bi-directional exchange of transactional and coordination occurs between mobile and processing units. In such a context, mobile units upload their pre-processed data for computing processes. These units can also transmit their vital information, such as the remaining power and current communication signal level, to a central facility for regular tracking purposes. Meanwhile, communications initiated by processing units include computing results, such as routing decision information or density information. Additionally, these units instruct new tracking waypoints or monitoring locations that can dynamically change according to a disaster event.

- *Processing-to-processing (P2P) units communications*: Collaboration signaling at main processing units mainly involves transactional data, including short-term computing data and long-term data archive. The immediate data produced are partial results of the algorithms executed in computing facilities. Meanwhile, the archive mainly contributes to future usage and analytics purposes.

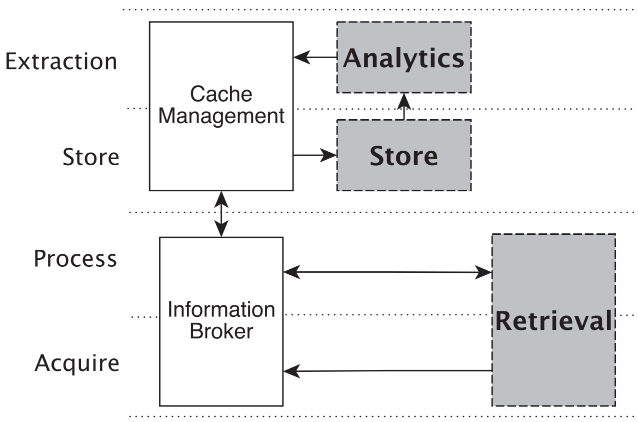


Figure 5: Diagrammatic processing blocks for Exchange and Communications functionality.

Looking at the above three communication patterns during the evacuation process, computing services such as fog infrastructure require a scalable communication capability. Herein the ability to carry out data transmission during increasing traffic is crucial. High-latency message dispatch and system failure are the two most common problems that must be avoided. Various input data types handled by FADE will add further complexity because the different types require different processing.

In light of the above-mentioned challenges, information broker aims to provide an integrated service of message handling and dispatch activities. This service is a one-stop support where the message is retrieved and subsequently propagated to relevant receivers. Alternatively, this functionality also can be seen as a middleware layer that bridges the data and communication management. Data exchange will also be configured to minimize the effort of various devices and communication management. The holistic concept used in middleware [119] is one approach that sees this pervasive device setup as an integrated facility. This principle proposed an abstraction layer that provides seamless access to the data transmission and persistent storage. In disaster management, this layer should inherit standard middleware specifications [119, 57], such as:

- *Interoperability*: A message handling service should provide a common interface to different data providers or types.
- *High throughput*: Transmission and reception of messages should be of low latency and minimally impact the process block. Synchronous data transfers where the sender waits for an end-point response should be avoided.
- *High reliability*: The service should be able to operate when needed without having an unexpected failure or a halt. This can be linked with an alternative backup procedure or a fault recovery capability.

5.1.1. Communication Scheme

One of the crucial aspects of the information broker functionality is setting up the communication pattern. From the OSI network layer perspective, a transmission setup can be categorized into unicast, multicast, or broadcast. The unicast pattern involves one-on-one data transmission. This delivery can be identified by looking into the packet source and destination address. Meanwhile, multicast and broadcast pattern disseminates data to several destinations simultaneously. The usage of these patterns in a natural disaster was highlighted in [6]. In such a setup, the communication among civilians used a unicast pathway, while the contacts between government and civilian use broadcast transmission. Despite this demonstration, each of these setups has its characteristics and should be carefully examined to accommodate various data provider's performance metrics, such as packet arrival time, length, category, and priority. As investigated in [120] and [121], multicast and broadcast can potentially deteriorate the system performance when the network is highly dense. These studies showed that unicast data transmission tends to have a lower delay and drop probability.

While the above-mentioned mechanisms focus on the network layer, Publish-subscribe (Pub/Sub) [56] is a high-level communication strategy that can simplify mass data dissemination. Compared to the three previously mentioned patterns, Pub/Sub does not require an explicit declaration of receiver address. Instead, the messages will be handled and disseminated by an event service. This setup is similar to multicast communications where two roles, producer and consumer, exchange messages according to its topic. The producer composes a message within its category, and then, an event service will propagate it to interested receivers. In order to receive messages, these end-points should specify and subscribe to one of the categories beforehand. Such a configuration can deliver multiple messages avoiding unnecessary message flooding.

Other benefits of Pub/Sub configuration are time, space, and synchronization decoupling [56]. Producer and consumer are loosely coupled in time because these actors are not necessarily connected all the time. In such a case, a producer can still emit messages while a consumer is offline. Pub/Sub also acts as a message proxy where producers and consumers do not know each other. Meanwhile, synchronization decoupling enables connectivity where a producer can continuously publish events and consumers can asynchronously retrieve it.

There is a diverse development of Pub/Sub services stemming from the notion of topic. Eugster *et al.* [56] specified three Pub/Sub variations, namely topic-based, content-based, and type-based. Content-based Pub/Sub requires subscribers to declare its interest on the actual contents of events. This subscription facilitates a subscriber to specifically configure event notification based on its intrinsic property rather than an explicit pre-determined label. Meanwhile, type-based Pub/Sub provides subscribers to access not only an event content but also its structure. Pub/Sub usage was mentioned in [119] as event-based middleware but it is only limited to a topic-based subscription. In fact, its implementation can be extended to a core communication technology in the service-oriented middleware. Not only can Pub/Sub dispatch a message to end-point devices, but also it propagates a coordination message to an underlying element, such as a storage service or a Quality-of-Service (QoS) management.

5.1.2. Networking Architecture

Infrastructure plays an important role to support information broker functionality with an increasing data traffic. Typical setup, host-to-host IP-based transmission, mainly considers packet delivery based on the device address, and it is difficult to catch up with the content-oriented delivery [122, 123]. The performance of a conventional communication setup often suffers from high data traffic produced by heterogeneous devices [124]. Traffic engineering with network functionalities decoupling [57, 125] and alternative IP-based networking [122, 123] have been envisioned and increasingly popular to tackle such challenges. In the following parts, we discuss the concepts and the FADE usage of two emerging networking capabilities, namely software-defined networking and information-centric networking.

Software-Defined Networking (SDN) enables a separate implementation of three key components in network switching, i.e., control, data, and management/application planes. In contrast to the conventional networking setups, the SDN main logic unit can be deployed at a centralized facility or distributed locations. The control plane contains a logic processor and two configuration interfaces, namely North-bound and South-bound Application Programming Interfaces (API). The North-bound API connects the upper layer, application, with the control plane via a high-level user interface. This API typically helps users to configure network policies and management, such as QoS, routing, monitoring, and traffic engineering [125]. On the other hand, the South-bound API bridges the coordination between the control and the third layer, data plane. In this API, the technical device-related configuration is delivered from the network intelligence component to data switching devices. The instruction for these middleboxes is commonly written in OpenFlow (OF) syntaxes in order to support interoperability across vendors.

The availability of programming interface in SDN offers flexibility in network management, thus making it able to support a higher network functionality, such as fine-grained access control, traffic engineering, and network virtualization [58]. Furthermore, SDN is envisaged to enhance traffic engineering in four areas, such as flow management, fault tolerance, topology update, and traffic analysis [125]. In a typical SDN implementation, network policy can be written in high-level languages [59, 60, 61] or logic-arithmetic expressions [62], which are then translated into OpenFlow instructions. Among these languages, Pyretic[61] enables modularity in SDN by its policies-as-functions approach. Due to this capability, SDN has been extensively applied to highly-demanding networks, such as urban sensing [46, 57, 126] and VANET [127].

Considering the implication of SDN on various network management, several possible adoptions to FADE are listed in Table 5. Firstly, the device group control in SDN can enable network devices to be categorized and aggregated based on the device types. Motivated by the two distinct interfaces in SDN, namely the Northbound and Southbound, the authors in [126] conceptualized mapping of physical network entities, intelligent capabilities, and high-level application access. The study suggested a dedicated controller based on the network device group, and then applied optimization computation at each controller. Similarly, reference [127] devised a VANET application in which the term "layer" is a reminiscent of an SDN component. In this study, an ITS system composed of wireless nodes, Road-Side Units (RSUs), and Road-Side Unit Controller (RSUCs) maps their functionalities into the data plane and controller of the SDN.

The second possible adoption is an SDN-based middleware service as proposed in [57, 63]. SPF, the abbreviation of *Sieve, Process, Forward*, [57] was originally motivated by the similarity property between the IoT and SDN concepts, namely simultaneous information processing and data dissemination. In this framework, an IoT application is described as a service that is accessible on-demand and managed with priority levels. The framework replaces the original functionality of Open Networking's SDN architecture, i.e., the Data Plane, with a middleware-inspired principle, i.e., Information Processing and Dissemination Plane. This new functionality that resides in the SDN's Programmable Gateway is controlled by SPF instructions and is responsible for processing messages based on its priority, and tags through the Information Processor. The processed messages are optionally applied content-wise filtering, usually referred to as *Sieve*. A collection of data-processing pipelines further process more complex computation, such as video processing and Optical Character Recognition (OCR). This step is called *Process*. The final step, i.e., *Forward*, disseminates the content to relevant parties. Reflecting to this study, FADE can adopt the SPF capability to circulate on-field evacuation data based on using early processing via an SDN's programmable component. Thus, a lower latency data exchange can be achieved since the computation load is spread throughout the network components.

Lastly, the third viable endorsement is an SDN-based Pub/Sub communication pattern. Briefly discussed in Section 5.1.1, Pub/Sub is a data transmission strategy that allows some entities for producing information with several topics and letting other parties to consume messages with preferred topics. Not only does this mechanism reduce the complexities in the implementation process, but also it offers high potential to improve applications' responsiveness to certain events [64]. PLEROMA [64] exploits this undeveloped improvement with an SDN-specific Pub/Sub configuration. To this end, the network flow updates should be constrained to latency efficiency, bandwidth usage, and cost efficiency. A content-based subscription

Table 5: State-of-the-art SDN-based Technologies

Reference	Aspect	Possible FADE Adoption(s)
[126, 127]	Device group control	Data plane for each physical devices
		Categorical grouping of physical devices
		Unique control plane for each device group
[57, 63]	Middleware service	Configurable and programmable gateways at data plane
		Service-based level prioritization
		Unique dispatch channel with parameter specification
[64, 65, 66]	Pub/Sub dissemination	Pub/Sub message push from data plane
		Multiple optimum independent Pub/Sub flows

model is used in this algorithm to achieve this aim. The model consists of a set of attribute pairs referred to as possible events which publishers disseminate. Then, through a subspace relationship between publisher and subscription/advertisement, a packet with a certain header and length is disseminated as an event. The controller must install the flow rules on each switch along the path between the publisher and the subscriber. Meanwhile, the authors in [65] integrated an SDN component with another Pub/Sub middleware, namely Object Management Data Group's Distribution Service (DDS). The algorithm allows programmable data plane to have the mobility management, the dynamic channel configuration, and the rapid client association.

Information-centric Networking (ICN) is one of the future networking projects that include various sub-project milestones, such as Named-data Networking (NDN), Content-centric Networking (CCN) [123], and others cited in [122]. This project mainly aims to focus more on information transmission rather than end-point communications. ICN prefers multicast mechanisms and in-network storage by labeling the message content in the network layer.

The ICN concept is an ideal alternative for a data-exchange functionality of FADE where on-field information itself is more important than the source of data. One of the ICN properties, i.e., processing and forwarding can be based on the message interest. The transmission will no longer require to mention the destination address as replaced by the device interest [123].

5.2. Cache Management

Cache management is another important component in data exchange due to the high-volume data circulation as a product of information transmission and analytics features. This is similar to Big Data subject where massive live raw data are processed and prepared by a real-time messaging service [128]. Then, these data are passed to stream processing platforms for further operations. Due to the vast amount of incoming data and low-latency output requirement, a highly-durable and fault-tolerant data processor is required. Motivated by this existing field implementation, cache management in FADE should be robust and scalable to support the the short-term computation. For this purpose, a distributed streaming platform is powerful to handle the high data load. The reliability of this technology was investigated in [68] where three primary platforms were run under billions of records. The authors focused on two main performance metrics, such as throughput and latency, using proposed penalty factor formula.

In FADE, cache management does not only handle streaming processing for transactional data, such as sensor data or people's movement, but also manage long-term data maintenance. For this reason, this functionality requires a scalable data storage solution that can process a high input data traffic. A NoSQL solution, e.g., MongoDB and others cited in [67] offers unstructured data storing due to their reduced effort with the data modification characteristics. Besides, it requires a lower cost because of its "scale-out" scheme. NoSQL-based data management such as Redis, MongoDB, Cassandra [69] and HBase [70] can process high data inputs leveraging a distributed storage and processing scheme over multiple storage. Similarly, FADE can also utilize a distributed streaming platform, namely Spark [71], Samza [72], Kafka [73], and Storm [74] for cache management purposes.

6. Intelligence Functionality in FADE

Intelligence functionality mainly handles all the important calculation processes in FADE, which include routing decision algorithms and system metric optimization. The routing mechanism primarily concerns how to identify and allocate evacuees using the available resources. Meanwhile, the metric optimization governs how computing components are configured to achieve optimal computation goals. Fig. 6 illustrates the interaction between the sub-elements within Intelligence functionality.

6.1. Metrics Processing

System measurement and configuration requires a careful attention since a high volume of data and computation tasks is done during an evacuation process. Measurement will determine how well the current resources can serve the current requests while the configuration is a set of actions that are executed to manage computing facilities to achieve the goals. These goals are usually related to QoS, which commonly mentions various performance metrics [129]. In FADE, we mainly highlight the following metrics.

- **Throughput**: The maximum number of valid responses that can be served by a system.
- **Response time**: The time difference between the request sent and the system response received.
- **Availability**: The amount of time that a system can provide services with an expected performance level.

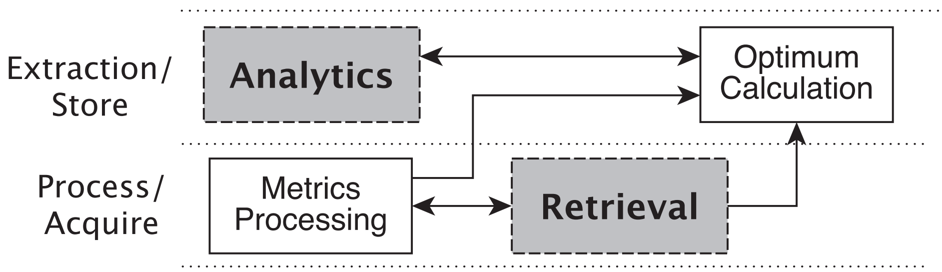


Figure 6: Processing within Intelligence functionality.

Kashani *et al.* [129] found that the system-wide QoS performance can be solved by different approaches. According to the study, QoS-management can be classified into three groups, namely communication, resource/service, and application. While the communication management maintains the network resource, the service management concerns about the governance in computing. Meanwhile, the third group exploits the fog configuration so that certain applications can work and satisfy the QoS requirement. Among these three categories, the first two are the most relevant approach to FADE. We will elaborate the discussion on this classification to match the utilization in FADE in Sections 6.1.1 to 6.1.2.

6.1.1. Network management in FADE

In FADE, network conditions, including M2M and M2P communications, are measured and managed carefully. M2M transmission management includes the transmission of data captures and message coordination. These types of communication require a minimum delay and a high packet delivery rate. Meanwhile, M2P data exchange covers the data transmission of M2M and decision. This data exchange can utilize delay-sensitive or minimum-latency algorithms that leverage transport [75, 76, 77, 78, 79, 80, 81] or application [82, 83, 79, 84, 85] layers.

One approach for optimizing the final transmission time is by minimizing the hop-to-hop delay, which is achievable by improving the default transport-layer mechanism [75]. In [75], the authors designed a multicast protocol for multi-rate MANET that considers both the end-to-end and one-hop transmission times on its forwarding table. The study focused on how to obtain the optimal sum of transmission time by minimizing neighboring devices blocking time caused by a high number of accesses. Meanwhile, the work in [76] investigated the caching mechanism effect on the network capacity. The strategy shows that different popularity levels of circulated packets affect the distribution model. Such a condition requires a caching strategy so that the network capacity can be further improved.

The second feasible method to optimize the end-to-end transmission delay time is by creating high-level rules or policies. In such a case, algorithms are applied to all IoT devices to manage how data are circulated. Reference [84] focused on the availability aspect by improving the system network lifetime. The IoT network lifetime is one of the crucial issues since the device has a limited amount of energy resources. In that work, the authors proposed a mechanism to form an inter-cluster cooperation within the IoT network to minimize the energy spent on the transmission, which in turn contributes to the system availability. Similarly, the work in [83] designed an algorithm for task scheduling in IoT that considers the device mobility and the task execution. The study formulated mixed-integer linear programming (MILP), which was then solved using both the offline and online strategies.

6.1.2. Service management in FADE

Service management mainly concerns with the throughput and availability of the core system. Expensive computation done at the processing unit requires careful management so that the QoS requirement can be fulfilled. To that end, various strategies to maintain a proper working system can be used, including resource allocation [130, 25, 26, 131, 132, 133], task scheduling [134, 135, 136], or provisioning [137, 138, 139].

Herein resource allocation determines how computation should be carried out either locally or shared with Fog Nodes (FN). Most of the strategies listed in Table 6 evaluate the efficiency of the offloading process by considering various objective and constraint metrics. A study by [131] considered a three-tier computing scheme where IoT, Fog and Edge devices can contribute to complex and demanding requests. This work devised an algorithm that adaptively allocate the optimum computing load to a tier of the system. The principle of this algorithm is calculating the unit slot allocation to achieve the minimum application loss and delay. It then accommodates non-uniform cloud capabilities by specifying the average processing rate and time. The strategy utilizes Lyapunov optimization to solve task allocation considering a three-way tradeoff among the average response time, the average cost, and the average of application loss. Similarly, Liu *et al.* [132] used a weighted approach to solve a non-linear problem in a three-tier topology. It then automatically arranged cloud offloading whenever the request rate is higher than the fog capability.

Table 6: Offloading Optimization Strategies in Fog Environment

Source	Metrics		Target (node)	Comm. model	Topology (Tier)	Formulation
	Objective(s)	Constraint(s)				
[130]	energy, latency	rate, deadline	Single	OMA	2	NLP
[25]	energy	power	System	NOMA	2	MINLP
[26]	energy, latency	latency, power	Single	N/A	2	NLP
[131]	latency, cost, loss	latency, loss	System	N/A	3	NLP
[132]	energy, latency, cost	rate, deadline	Single	NOMA	3	NLP
[136]	throughput, task completion	latency, capacity	System	N/A	3	MINLP
[133]	energy, latency	latency	System	N/A	Fog-fog	NLP

Note. NLP = Non-linear Programming; MINLP = Mixed-Integer NLP

A slightly different scenario was used in [130, 25, 26] where a two-tier topology was considered as a computation offloading scheme. Reference [130] tried to minimize each offloading decision by looking into the available (idle) CPU as well as the proportion of CPU usage if the task is being offloaded. Meanwhile, an algorithm to decide whether to offload a certain task (partitioned workload) to the fog units was devised in [25]. It attempted to minimize the average of the system energy usage using an improved genetic algorithm that approximately solves an NP-hard allocation problem. In contrast to [88, 25], the work in [26] introduced a fairness factor to evaluate task allocation between FNs. In this scenario, one client was associated with one FN, and the client could choose which FN is suitable for offloading.

Task scheduling strategies have been studied in the fog environment to manage task execution. Li *et al.* [134] designed a scheduling strategy based on deep-reinforcement learning to solve a hierarchical fog structure. The meta-heuristic learning-based solution offers flexible scheduling as it was not specifically designed for one topology. Meanwhile, a genetic algorithm-based solution was proposed in [25] to solve a mixed-integer nonlinear programming problem of Non-orthogonal Multiple Access (NOMA) in the fog environment. Ni *et al.* [135] devised a method of Petri net-based dynamic-task handover between fog-cloud system. The strategy considers price and time cost of computation tasks as input for predicting task completion time. The result is then used as input for a dynamic resource allocation algorithm.

Provisioning as a type of potential service management in FADE mainly aims to determine the optimum decision of an application deployment as a multi-tier setup [137, 139] to minimize the cost and meet the QoS requirement [138]. In [137], dynamic placement can be formulated as a non-linear programming problem and solved using a greedy algorithm. Similarly, Yao *et al.* [138] formulated a deployment problem as a multi-objective optimization problem and solved it using an approximation method. A slightly different strategy was used in [139] where the problem was solved using two-stage algorithms. The first step provided a possible deterministic solution while the second one obtained the final solution using a heuristic approach.

6.2. Optimum Calculation

6.2.1. Crowd counting

Crowd counting is an activity of object detection and counting from an image at a certain crowd scene [112]. This work shows that this technique is pre-requisite for a crowd analysis where the number of people and density on a single scene can be estimated. Many applications have applied this technique for critical missions [140, 141] and city planning [142, 143]. As a solution to which a computing facility for evacuation is dynamically deployable and scalable, this capability is a key to estimate the road density and allocate evacuees to a certain part of the network to avoid traffic congestion. There are two main categories of crowd counting according to [112], namely conventional- and CNN-based.

The conventional strategy includes three approaches [112], i.e., detection-based counting [86], regression-based counting [87, 88, 89], and density estimation-based counting[90, 91]. In a detection-based approach, an object is recognized by its parts, such as head-shoulder [86] or as a whole [48, 144, 145, 146]. Fast whole-human detection is mainly enabled by the unique descriptors extracted from the human data [112]. This approach, however, is unable to perform well under conditions of high-density crowds and high background-clutters [93]. Meanwhile, Regression-based algorithms [87, 88, 89] obtain features mapping from a local image patch after completing both the global and local feature extractions. Meanwhile, the detection-based approach attempts to avoid the hard task of learning and localization by incorporating spatial information. Such information is obtainable from a learning process between local patch features and object density maps.

The CNN (Convolutional Neural Network), commonly known as ConvNet [92], is a method that utilizes learning network capability with the input of image patches or the whole images. As one of the deep learning technologies, learning is done by taking a lot of raw data inputs and letting the algorithm discover the required model for detection or representation [92, 93]. This is enabled by trained convolutional weights [147]. The algorithm is designed to process various data arrays of which size can be used for different processing purposes [92]. For instance, a 1D array is used for signal processing; a 2D array is required for image/audio processing purpose; and a 3D array is applicable for video processing. This algorithm takes benefits of the properties in natural signals, such as local connections, shared weights, pooling, and the use of many layers [92]. Because of this robustness, the algorithm has been adopted in a wide range of applications, such as object detection [147, 93, 148], classification[94, 1], and face detection [95].

A convolutional layer can be formally denoted by $X_i = W_i \otimes [X_{i-1}, 1]^T$ where X_i is the feature map of the input and the output of the i^{th} layer, $W_i = [W_{i1}, W_{i2}, \ldots, W_{ik}, b_i]$ corresponds to filter parameters of the i^{th} layer with b_i acting as bias and $\otimes$ denoting a convolution operation [93]. The fully-connected layer was obtained in [93] by $y_{im} = W_{im}X_{(i-1)} + b_m$ where W_{im} represents the m^{th} filter of the i^{th} layer. In an image processing application, a CNN-based algorithm typically uses interconnected processing layers. There are two types of layers used for this purpose, namely convolutional and pooling. As described in [92], the convolutional layer is organized into feature maps where each unit is linked to local patches in the feature maps of the previous layer via a set of weights called a filter bank. Then, the output of this layer becomes the input of the non-linear layer. The units that belong to a feature map share the same filter bank. Different feature maps within a layer use different filter banks. The main reason behind such arrangement is local groups of values in the image data that are likely to correlate and the local statistics invariant to location. Then, the convolution method using the feature map is applied on the filtering operation. Because the filter processing in each layer produces a higher dimension, the pooling layer reduces the feature map based on the operation of local patch data, which are typically maximum (max-pool). Commonly, a CNN architecture utilizes two or three stacked stages of convolution, non-linearity, and pooling [92, 147].

The CNN-based approach in [112] was grouped by model setup into basic and advanced techniques. According to this study, basic CNN [147, 93] incorporates a default functionality of CNN stages in terms of layers. Such a setup utilizes the basic setup of a convolutional network to detect the number of object given data input. Meanwhile, advanced CNN algorithms configure their layers to improve the network's robustness, lower error results, or combine the crowd counting process with additional tasks, such as foreground-background separation and crowd speed estimation. To achieve these objectives, these algorithms utilized various resolutions or scales [149, 150], local-global contextual information [151, 152], or multiple factors, such as deep-shallow network combinations [153].

6.2.2. Routing management

Massive evacuation during disaster events from the affected population to a shelter or safe area [1, 3] requires a careful planning and execution due to the presence of a huge number of road usages [4]. The developed plan should consider randomness in factors that might affect the evacuation support, such as the evacuees' compliance, the rate of evacuation, the unexpected traffic loads, and the current road network condition [4]. As described in this study, a pre-computed evacuation plan, such as [1, 3], is less-realistic and less-feasible due to issues such as the possibility of congested recommendation route, the growing problem-complexity, and the less-detailed assumption in the model. To cope with these limitations, online or or-demand evacuation systems, such as [8, 4], were proposed. These studies offered active responses for an ongoing evacuation process by providing scalable communication platforms [8] or a real-time road management [4] integrated with Intelligent Traffic System (ITS).

This sub-functionality of FADE has the main responsibility to decide the optimum route for each group of evacuees using relevant transportation management strategies, such as trip assessment [154, 155, 156, 157, 158], traffic assignment [159, 160], or flow prediction [161, 162, 163, 164]. The routing management is one of the crucial aspects to mitigate traffic congestion during a disaster as the road network capacity and demands can drastically change over time [4]. Since the road network is composed of smaller road sections, the aforementioned management schemes can give additional supports in terms of traveling time variability and mobility profile. Compared to [8], FADE aims to provide an online or on-demand solution with less dependency on the availability of ITS. To achieve such a goal, as mentioned in Section 4, some UAVs can be dispatched to different parts of the affected population. These units will act as an evacuation assistance with various degrees of involvement depending on the hazard levels. The usage of multiple hazard event models [165, 166] in the deployment plan will improve the UAVs dispatch decision.

7. Analytics Functionality in FADE

Analytics process of FADE (Fig. 7) can take place on different tier of the system, e.g. cloud and/or fog units, where computing resources are vastly available. In Analytics functionality, computed solution from Section 6.2 or stored trajectory-data are computed either to dynamically configure the subsystems of FADE or predict upcoming request characteristics. Generally, the enabling technology in this functionality is algorithms with forecasting or prediction capabilities.

7.1. Reconfiguration

Reconfiguration is a key activity that enables the infrastructure adjustment under a highly dynamic environment. Fluctuation of the incoming data or other coordination messages can change rapidly depending on the current evacuation conditions. As an example, road destruction or blockage during a multi-hazard disaster event [165] can result in the evacuation re-routing.

Table 7: State-of-the-art Normal Traffic Management Summary

Aim	Reference	Main concept(s) used
Network quality assessment	[154]	New maximum equilibrium model proposal for dynamic OD reconfiguration.
	[155]	Road partitioning using α-Cut on density peak graph (DPG) produced from clustering and link aggregation.
	[156]	Correlation analysis between space and time using proposed function to discover local and global measure.
	[157]	Evaluation on network scalability using TSI and NAI indicators via proposed large-scale high-frequency data processing model.
	[158]	Hybrid algorithm implementation on tripartite graph to rank node importance.
Traffic assignment	[159]	Traffic flow regulation based on daily-basis stochastic model.
	[160]	Traffic light scheduling system using proposed controlling algorithm.
Time-basis traffic load prediction	[161]	Two-stage prediction framework using spatio-temporal variable derived from VS-SVR model.
	[164, 163]	Traffic forecasting using Deep Learning on transformed directed-graph traffic-flow
	[162]	Day-to-day traffic flow estimation using statistical method.

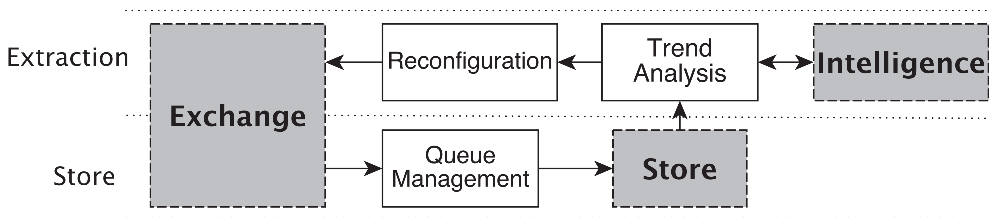


Figure 7: Processing diagram for Analytics functionality.

After optimizing the traffic assignment using one of the approaches in Section 6.2.2, some evacuation groups can join another cluster on the rerouted path. As a result, a network surge will be likely to happen because more people are being monitored and UAVs coordination will be more intensive. Alternatively, the network utilization can also be lower whenever an evacuation group could reach a safe point. This increase or decrease should be monitored and configured so that energy or computing resources can be improved. Although a scalable data exchange in Section 5.1 has been carefully designed, human intervention might not keep up with unexpected changes. Improvement in the transmission control and traffic engineering that leverages intelligent capability should be considered to assist human execution.

7.1.1. Congestion Control

Congestion control is a way to manage the network usage so that its performance can be maintained during any circumstances. It was proposed as a response of internet meltdown in a few decades ago by providing a back-off mechanism whenever a congested transmission occurred [167]. It is also one of the necessary and powerful features [167] in the network because it manages the number of packets transmitted [104]. However, its implementation is still limited in certain scenarios and requires a further development for intermediary devices [167, 168]. Congestion control implementation on end-devices tends to be a reactive approach, which means that the proposed strategy works after the occurrence of congestion. As initially specified in [167], two additional congestion controls at router devices were proposed, namely queue management and scheduling algorithms. Queue management controls the packet queue length by dropping packets whenever necessary or appropriate. On the other hand, the scheduling algorithms manage the order of packets transmission and are mainly used to manage the allocation of bandwidth among flows. Furthermore, according to the aforementioned work, typical queue management applies a "tail drop" strategy where packets are received and kept in the buffer until its full capacity, and dropped if it reaches enough capacity.

RED [169], an Active Queue Management (AQM)-based strategy, was formulated to address the conventional strategy in estimating the average queue size and determining a packet decision. Since then, more and more research was conducted to improve the AQM [170, 168]. The work in [170] classified previously proposed strategies into two groups, namely end-system-based and router-based. Router-based strategies run and process the queue at a router-level while the end-system-based algorithms aim at the end-to-end flow definition and implement it at the end-devices. Between these two, FADE is more suitable to the second option as the decision whether to use flow and control or not is determined by the end-devices. The purpose of queue management in this study is to provide fairness at some relaxed degrees to allow unresponsive flows to gain more bandwidth as long as they do not deteriorate others. Strategies such as LQD [96], BRED [97], FRED [98], SRED [99], CHOKe [100], SFB [101], BLACK [102] and CARE [103] can be further realized by the programmability feature of the next-generation network infrastructures, such as SDN or NFV [168].

The aforementioned router-based algorithms were developed based on the heuristic principle of which parameters should be tuned manually. As a consequence, the produced action does not adequately represent time-varying and non-linear network conditions [171], and can thus quickly result in system instability. Some algorithms [98, 103] might have limitations on the scalability due to its per-flow information and sample capture [170]. Furthermore, according to [170], the techniques in [102, 100, 101] have the potential of penalty mismanagement due to a simplified or absence of flow information. An alternative to these heuristic concepts is neuron-based AQM strategies that offer solutions that can not be solved by conventional rule-based programming [171]. In addition, this work suggested that algorithms such as NN-RED [172], Neuron PID [173], NRL [174] and AN-AQM [175], have provided various improvement to heuristic-based in terms of adaptability, behavior transparency and stability.

7.1.2. Network Prediction

Machine learning (ML) usage in network [104, 105] has enabled diverse management capabilities to assist or automate human involvement. According to [104], one possible implementation of ML in network is traffic prediction, which can be either a pure Time Series Forecasting (TSF) or Non-TSF problem. While TSF strategies build a regression model capable of drawing accurate time-to-time correlation traffic volumes, Non-TSF predicts traffic using various methods and features [104]. Between these two classifications, ML utilization for the FADE network is more related to the Non-TSF problem where traffic is likely to be unpredictable and dynamically evolve depending on the evacuation progress.

A method based on the frequency domain was used for the network flow analysis in [20]. This work focused on predicting incoming and outgoing traffic volumes on the inter-data center link. Similarly, the work in [107] utilized a regression technique using Bayes Rule and RNN with LSTM on temporal dynamic information. The strategy aims to predict future traffic volumes based on the current flow count. Alternatively, an online Bayesian Moment Matching (oBMM) was used in [106] to predict the size of flow and to detect an elephant flow. The method considered various features, such as communicating device IP address and port, protocol, server versus client, and initial three packets size after establishing a connection. Detected elephant and mice flows offer further routing optimization where each flow can be allocated to a different path [106]. In such cases, mice and elephant flows were configured with equal-cost multi-path routing and least congested path, respectively.

7.2. Trend Analysis

In this part, the strategies for historical mobility data pre-processing aiming at evacuation modelling and transactional data post-processing are discussed. This section differs from Section 6.2.2 in that part mainly concerns with the optimal calculation using real-time data. The preparation activity objectives of trend analysis are to obtain daily travel models of a local citizen, typical traveling time, and critical nodes that often cause traffic congestion. Meanwhile, the post-processing task manages, discovers, and learns exchanged data during and after evacuation. The learning and discovery activity improves variables and models to better represent the affected population. Due to the absence of the region-specific citizen behavior, this activity will help improve prediction capability for future usage.

Trajectory mining offers a wide range of capabilities on managing mobility data, such as derivation, pre-processing, management, and other higher-level tasks [41]. According to the study, the high-level task in mining activity includes pattern mining, outlier detection, and classification. The usage of trajectory mining was applied to reveal valuable trip knowledge, such as destination prediction [16, 17, 18, 19, 20], point-of-interest discovery [40, 108], and route optimization [176]. To this end, researchers have exploited various inputs, including Spatio-temporal [20] and semantic/annotated [16, 108] information using stochastic [17, 20, 177] to learning [176, 18] techniques. The following section will categorize the analysis in details and review relevant studies.

7.2.1. Trajectory Formulation

The initial step in the analysis is a model formulation which leads to different subsequent processes. The model construction may utilize existing data such as stored timestamped-geospatial data provided from a communication trace [41], or a visual capture [178]. Data with time and location information highly represent traveling situations as people change positions over time. Such data representation is also called as Spatio-temporal model. Another trajectory representation, Social model, is formulated based on labelled human interaction records derived from real-life analysis.

Table 8: Summary of Trajectory Analysis strategies.

Ref.	Aim(s)	Primary Technique(s) used
[16]	To predict trajectory destination given collected trajectory data and temporal sensitivity issue.	A clustering algorithm applied on a semantic-based probability model.
[17]	To improve trajectory prediction result given a sub-trajectory query.	A two-dimensional Gaussian mixture-based prediction model.
[18]	To estimate a travel destination given various external features.	An improved Recurrent Neural Network technique applied on Deep Learning.
[19]	To predict the destination of a sub-trajectory query considering time sensitivity.	A Markov model with a time-based transition state.
[20]	To improve trajectory prediction result given incomplete/insufficient collected trajectory data and temporal sensitivity issue.	A Markov chain-based learning model and a derived tensor-based prediction model.
[40]	To group travel destination spots based on their popularity.	A clustering algorithm applied on a trajectory partition with a stable stationary point.
[108]	To discover hidden factors from given time-interval travel.	A similarity measurement method applied on filtered sub-trajectory data.
[176]	To discover travel pattern, and to estimate future travel based on known regular routes.	A probability-based movement pattern matching technique applied on a constructed pattern tree.

A spatio-temporal model records trajectory information as a point-to-point location in a chronological order, which creates sequential information illustrating people's movement. Each entry of this model lists visited locations along with their time duration. A simple model using directed graph was defined in [41] where each pair of starting and termination point is connected with an arrow. The connector is then labelled with a number indicating the travel duration. Meanwhile, in [17], a tuple-based model is used to record all objects position at each time interval. Then, its time assumption is eliminated to solve a limitation on trajectory sampling during two consecutive times, and to produce a *piece-wise* segment.

A social model considers additional metrics derived from preference / semantic labeling [108, 179, 16] or a social interaction score [180]. In [108], a set of categories and its factors determining human decision were listed. Then, each factor's significance was calculated using linear model estimation. Finally, the basic utility function was estimated using a linear regression technique to estimate each category's significant factors under a certain error rate. Meanwhile, a variation of this model, the Lifestyle-based Trajectory Model, was devised in [180] by taking into account three principal components, namely topic-specific location generation distribution, lifestyle-specific topic transition distribution, and user-specific lifestyle selection distribution. Consequently, the model can derive useful information, such as the probability of a topic being produced at a certain location and the user's degree preference over a lifestyle.

7.2.2. Pattern Processing

The next step in analytics is model processing that aims to cluster trajectory or predict the trip destination. Trajectory clustering aims to seek the representation of common paths shared by different moving objects [41]. Meanwhile, trip destination prediction resolves the end-point of trajectory given a certain input. FADE makes use of these strategies for two goals, namely to group people's moving behaviors around the affected location so that the traffic management can avoid congestion and estimate how many trips are made to a certain critical location. To achieve these goals, previous studies proposed different strategies, such as stochastic model or process [177, 17, 20], graph clustering [181] and learning algorithms [18].

The similarity of multiple trajectories for a grouping purpose was investigated in [181]. The study tried to gain hidden information that exploits close spatial and temporal aspects of several trajectories. The motivation behind this clue finding was that some data points in multiple trajectories could be different in terms of time and position but actually, these trajectories capture the mutual behavior of a user. For this particular reason, the study initially calculated a similarity score between trips then generated a clue graph based on the rating. Finally, the study did a graph matching to discover trajectory clusters.

A Gaussian mixture model-based strategy was used in [17] to classify historical trip data and, subsequently, to predict the final destination. The algorithm initially calculates two scores and determines which trajectory belongs to which cluster. Then, based on the clustering map, the final destination of a query is determined by calculating a weighted sum of the mean final destination of each cluster.

Alternatively, a learning algorithm to determine the final destination was devised in [18]. The study offers a novel Bi-directional LSTM strategy, namely TALL, to accommodate the latent feature of preceding and following location given trajectory data. The usage of the learning algorithm was motivated by the fact that the Bayes rule-based strategies have accuracy limitations on sparse trajectory data and smaller granularity grids.

8. Input Acquisition and Extract Functionality in FADE

Data acquisition and extraction of raw image and sensing data requires careful treatment due to several factors. Raw data transmission during a high-demand access scenario will likely to suffer from latency and drop rate. As a consequence, devices will need to retransmit, which can cause more access to a communication channel at the cost of extra energy expenditure. In fact, not all sensing nodes or assistant units can afford this process because of the energy left. Therefore, some adjustments on the data process and extraction are required to manage these trade-offs properly.

A more complex computation is required within this functionality to minimize the system-wide energy, communication, and computation spending. Rather than separating task execution at a certain network level, mobile and fog units should share the proportional amount of jobs. For instance, data acquisition includes not only capture and transmission at the edge devices but also error checking and reception in the core network. In this case, the capture and error checking can be done at both levels. Data processing and extraction spanning from all the preparatory steps to the partial process should be done at both ends. The following subsection further explains the required adjustments to improve the trade-offs.

8.1. Component Adjustment

Input transformation includes process decomposition that distributes the acquisition and extract process to the edge and core network units. Referring to the offloading mechanism discussed in Section 4.2, FADE should retrieve input as information rather than a raw form using an offloading mechanism. Since crowd-analysis is a vital task in the evacuation management, the acquire and process functionality can be specifically designed for this purpose. The CNN-based algorithms as cited in [112] compose of various depths of convolutional, pooling, and max-layers. Rather than assigning computation tasks to mobile fog nodes, mobile devices should be allocated some parts of the task as demonstrated in [114, 116, 115]. This will result in a lower resource usage and queue at the main computing facility.

Flow management can mitigate the traffic congestion by the transmission timing management and traffic classification. In the timing management, the input can be acquired periodically or tentatively using a frame differencing algorithm as proposed in [117]. Without this setup, rapid retransmission will potentially deteriorate the network performance due to channel usage and gateways flooding. Additionally, the buffer capacity at the other side of the network is limited for receiving transmitted data from all units. Although the middleware capability mentioned in Section 5, such as message broker [73, 182], was carefully benchmarked in [183], there is no estimation of capacity or bottleneck. Meanwhile, traffic classification collects the network traffic and classifies them into different groups using SDN [184, 185, 58].

In-network computation offers a complex task to be offloaded at the network layer with three implementation prerequisites, i.e., it must reduce traffic significantly, have minimal application changes, and guarantee the result correctness [60]. This concept is typically facilitated by network programming implemented within the SDN. Due to its dynamic policy updates resulting from an event-based response [58], SDN is considered as a plausible option to perform complex computation jobs, such as distributed applications, caching distributed key-value stores, network diagnostics, and aggregation functions in data-centric processing [185]. Even some machine learning and graph analytics can run at the network level [60].

8.2. Stateful and Event-driven Network Programming

SDN facilitates more advanced network management through a programming script generated into flow rules. Despite all strengths mentioned in Section 5.1.2, the absence of a global variable representing the network conditions has resulted in several flaws in the SDN implementation. Firstly, due to the delay in rebuilding flow tables, mistreatment on processed packets is likely to occur. Secondly, any requests received by the switching devices will be forwarded to and hold at the controller during the rebuilding process.

The event-driven networking that is facilitated by the stateful programming language has tackled the traditional SDN limitation by providing a persistent global array [186, 58]. This information container gives access to flow-associated information on the topology and possibly applies rules resulting from a particular event. However, not all stateful network programming languages can guarantee consistency during the dynamic updates, react to a particular event at the data plane level, or update the protocol automatically [58]. Event-driven networking, i.e. SNetKAT [58], constructs a Network Event Structure that can hold consistency network properties during the transition between two network configurations.

Network-level computation facilitated by one of the network programming languages, i.e., P4, was demonstrated in [60]. Herein the DAIET proposal [60] was motivated by a possibility to do aggregation independently with several caveats. Firstly, the available resources, such as computation power and storage capacity, are limited. Secondly, it requires a specific application-network mapping where each step of computation must be associated with certain devices via flow rules. Despite these provisions, stateful programming can offer many opportunities for flow management via scripted instructions. Table 9 lists available stateful network programming languages with their known benefits and caveats.

9. Open Challenges and Directions

9.1. Exchange and Communication

Machine learning strategy on Traffic Management could be more utilized and tailored specifically for FADE based on certain requirements. This is mainly motivated by the fact that the characteristics of transmitted data are quite predictable. Trajectory patterns and frequencies of natural disasters in a certain location are unique, and, thus, can be learned over time. Therefore, a location- or disaster-specific machine learning using the pre-collected data

Table 9: Stateful Network Programming Language

Name	Strength	Caveats	Update	Event-driven?
Maple [187]	Network-wide forwarding behaviors	Non-reusable f function can potentially result in higher runtime.	$\mathcal{P}$	✗
	Policies are applied in conjunction with others	Difficult f function parallelization poses a challenge to implementation speedup.		
Merlin [188]	Constructs for wider scope of network management	Small runtime, called negotiators, is needed to accommodate dynamic policy update.	$\mathcal{P}$	✗
		Path constraint expression is available to tune scope matching flexibly.		
Flowlog [189]	Syntax is a reminiscent of SQL	Unclear network behavior during reconfiguration	$\mathcal{P}$	✓
		No loop and recursive construct is available.		
		Proactive compilation is performed occasionally.		
		Forwarding rules compilation output might not work when tuples refer to any external table sources.		
Stateful NetKAT [58]	Consistent network updates application with consistency properties hold during reconfguration.	Users needs to master the basic of Boolean and Kleene algebra expression.	$\mathcal{P}$	✓
SNAP [186]	High-level language	TCP policy rules addition might produce a surge of compilation time.	$\mathcal{R}$	✗
		Stateful operators may be partially compatible to some hardware.		
		MILP creation requires a longer time than the solution.		
P4CEP [185]	High portability due to P4 support	Synchronizing access to global state is not easy.	$\mathcal{R}$	✗
		Unable to directly handle global state of registers.		
		Loop construct is not available.		
		Action invocation is not a direct process.		

Note. $\mathcal{R}$ = reactive; $\mathcal{P}$ proactive flow updates is a viable option for optimized solution. Additionally, the image/text data transmission used in the system should have a different QoS setup to improve the network efficiency [11]. This motivates the optimization of supervised learning to train and test a network traffic model [105] adopted from large-scale evacuation scenarios [8], such as music concert, fair, and expo. After the model is generated, a testbed can be developed to see how well the traffic management handles massive traffic.

The ML usage on network reconfiguration can be further improved when combined with stateful [186, 189, 185, 187] and event-driven programming [58] in SDN. The learning process can predict the network current situation using input provided by these frameworks. Then, using the reactive rule compilation, the output of calculation taking in the form of metrics will be used to generate flow updates.

Transient connectivity during the evacuation process can risk computation latency due to the input retransmission. This is the case of MFUs [11] with limited computation and power resources. In this communication setup, MFUs are deployed within the transmission distance of MAUs and serve the dispatched MAUs. The communication quality between these two groups of units can be severely impacted by weather conditions and power outages of one communicating side. Future research can incorporate this parameter in designing the offloading decision algorithm so that a more realistic result can be obtained.

9.2. Computation Offloading

Unidentical resources of mobile units - CNN has been used in crowd counting and crowd-analysis and outperformed conventional strategies. If FADE implementation decides to streamline data preparation and intelligence, CNN computation can be offloaded hierarchically using strategies mentioned in Section 4.2.3. Dynamic-offloading learning strategies such as [115, 116] consider task partition only based on the hardware specifications, such as CPU or RAM of the computing units. However, this could risk a system failure due to non-uniform resource remaining that is caused by weather conditions, rerouted travels, or crowdsourcing approaches [8, 9]. It can also lead to a challenging situation for CNN-based crowd-analysis as mentioned in Section 6.2.1.

Inference layer partitioning was proposed in [115, 116] to distribute the learning process requests across end-devices and fog nodes. However, none of these studies have taken into account the remaining battery for model assignment. Each computing device may not be able to complete the task due to power outages. Learning model partition and computation on resource-limited devices such as MAUs should be implemented in real life for benchmarking purposes. The testbed on interactions between a certain computation layer and CPU usage [113] will help researchers finely tune model distribution and estimate the power usage.

Adaptive preprocessing configuration can assign independent tasks that compute the orthogonal models [115, 116]. However, the depth of these models should be defined based on two principle aspects: (1) performance metrics, e.g., latency, task completion time, and unit's computation capability, as computation objectives and (2) energy consumption as a constraint. The dynamic allocation under energy constraints is a critical task as it can prolong the system's lifetime and allow more effective responses to the disaster events. On the other hand, at the same time, relying on the minimized energy consumption might compromise the computation's accuracy. Therefore, it is also interesting to reformulate the problem as a trade-off between energy availability and computation capability. The existing algorithms provided in [114] seem to be a static invocation of such capability.

9.3. Learning Process

Model minimization using Binary Neural Network (BNN) [190, 191, 192] can reduce the computation required on the learning process. The BNN uses a 1-bit model, which has a proportional factor of 32x compared to the traditional 32-bit type. As a consequence, multiplication of the activation layer and 1-bit weight can be replaced by bit-wise operation, making the processing time less. Additionally, XNOR operation in the BNN has been able to expedite the neural network process. Even though BNN usage has been common in the edge-based learning process because of the complexity and space reduction, its implementation has been exclusively found in image classification and object detection. Therefore, further research is required to investigate its feasibility for crowd counting.

Application sub-steps mapping on SDN has enabled in-network computation [60] to speed up the application that uses simple arithmetic/logic operation and commutative/associative function. This work has shown some potential advantages, such as latency reduction, throughput increase on certain operations, and traffic decrease. Regardless of its promising results on machine learning and graph analytics [60], in-network computation requires a detailed mapping between each calculation step and middleboxes for rule flow updates. Future research can improve this mapping requirement by taking into account the One Big Switch approach used in [186, 189].

Supervised learning results of crowd counting depend heavily on the quality of the dataset despite the lower error rates [112]. The process of detecting and counting pedestrians on a single image requires a dataset that contains various density levels, no mislabeled samples, and generalization [112]. An adequate generalization could model the global context information that minimizes the overfitting in learning and increases the performance. In the case of a simplified learning network, e.g., AlexNet, the training data can be further improved by adding negative samples in which the ground truth count is set to zero [93]. This adaptation can improve the performance significantly up to 50% [93].

Outlier data detected from trajectory analysis could be used as an alternative path during the evacuation process. The fact that some people discover unusual ways is worth capturing. Compared to a typical mobility pattern, this unique path could promote a better traveling time under different circumstances. Therefore, the analytics part can store and prepare this data as a substitute option during the path computation process.

Algorithm-hardware alignment offers various potential benefits to the efficiency of an embedded learning process. As reported in [193], it can solve the memory and computational bottleneck, thus making it able to reduce the energy consumption. To that end, a learning algorithm should be enhanced or exploited by taking the hardware architecture into consideration. The inference process exhibits typical characteristics, such as specific data flow, robustness to approximations or fault introductions, and considerable sparsity [193]. The inference at separate locations [115, 114, 116] can be further improved by putting different types of processed data and using the specialized hardware. An energy-efficient platform with a good performance is critical for the inference process during disruptive scenarios.

10. Conclusion

Disaster evacuation guidance has shown as a powerful tool to mitigate the number of casualties as it can analyze traffic bottlenecks and output the optimum evacuation paths. The subject's importance becomes more critical as infrastructure is much less certain and available, but the demand increases suddenly, especially when a multi-hazard scenario occurs. Meanwhile, mobile fog infrastructure has been developed to support delay-sensitive applications through its localized service and scalable computing capability, making it a feasible option for authorities at natural disaster hotspots. Its scalability feature comes from a mixture of intelligent and resourceful computation, coming from diverse technologies. Motivated by the importance of evacuation guidance and the diverse element of mobile fog computing, this study has presented a comprehensive review of supporting technologies for a fog-enabled evacuation service under an umbrella called FADE. We have designed a top-down structure for logical functionalities that administers various technologies used in a fog environment, focusing on the state-of-the-art communication, computation mechanisms, and analytics capability to anticipate disruptive scenarios. Finally, this study has presented several research challenges spanning from intelligent data communication, offloading strategies to hidden information learning processes.

References

[1] H. Wang, A. Mostafizi, L. A. Cramer, D. Cox, H. Park, An agent-based model of a multimodal near-field tsunami evacuation: Decision-making and life safety, Transportation Research Part C: Emerging Technologies 64 (2016) 86–100. doi:10.1016/j.trc.2015.11.010.

[2] N. T. N. Anh, Z. J. Daniel, N. H. Du, A. Drogoul, V. D. An, A hybrid macro-micro pedestrians evacuation model to speed up simulation in road networks, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 7068 LNAI, 2012, pp. 371–383. doi:10.1007/978-3-642-27216-5_28.

[3] E. Mas, S. Koshimura, F. Imamura, A. Suppasri, A. Muhari, B. Adriano, Tsunami Evacuation Simulation - Case Studies for Tsunami Mitigation at Indonesia, Thailand and Japan, in: Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SCITEPRESS - Science and Technology Publications, 2014, pp. 249–254. doi:10.5220/0005104802490254.

[4] M. U. S. Khan, O. Khalid, Y. Huang, R. Ranjan, F. Zhang, J. Cao, B. Veeravalli, S. U. Khan, K. Li, A. Y. Zomaya, MacroServ: A Route Recommendation Service for Large-Scale Evacuations, IEEE Transactions on Services Computing 10 (4) (2017) 589–602. doi:10.1109/TSC.2015.2497241.

[5] J. Kim, S. Lee, S. Lee, An evacuation route choice model based on multi-agent simulation in order to prepare Tsunami disasters, Transportmetrica B 5 (4) (2017) 390–406. doi:10.1080/21680566.2016.1147002.

[6] P. Lieser, F. Alvarez, P. Gardner-Stephen, M. Hollick, D. Boehnstedt, Architecture for Responsive Emergency Communications Networks, Proceedings - 2017 Global Humanitarian Technology Conference (GHTC) (2017).

[7] F. Mezghani, N. Mitton, M. D. Francesco, A Multi-tier Communication Scheme for Drone-assisted Disaster Recovery Scenarios, 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) (2019) 1–7.

[8] A. J. Perez, S. Zeadally, A Communication Architecture for Crowd Management in Emergency and Disruptive Scenarios, IEEE Communications Magazine 57 (4) (2019) 54–60. doi:10.1109/MCOM.2019.1800626.

[9] M. Alwateer, S. W. Loke, N. Fernando, Enabling Drone Services: Drone Crowdsourcing and Drone Scripting, IEEE Access 7 (2019) 110035–110049. doi:10.1109/access.2019.2933234.

[10] Y. Kasai, M. Sasabe, S. Kasahara, Congestion-aware route selection in automatic evacuation guiding based on cooperation between evacuees and their mobile nodes, Eurasip Journal on Wireless Communications and Networking 2017 (1) (2017). doi:10.1186/s13638-017-0948-6.

[11] N. Fernando, S. W. Loke, I. Avazpour, F. F. Chen, A. B. Abkenar, A. Ibrahim, Opportunistic Fog for IoT: Challenges and Opportunities, IEEE Internet of Things Journal 6 (5) (2019) 8897–8910. doi:10.1109/JIOT.2019.2924182.

[12] T. Sakano, S. Kotabe, T. Komukai, Overview of Movable and Deployable ICT Resource Unit Architecture, NTT Technical Review Vol.8 (2010) 1–5.
URL https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201505fa1.html

[13] K. Katayama, H. Takahashi, N. Yokota, K. Sugiyasu, G. Kitagata, T. Kinoshita, An Effective Multi-UAVs-Based Evacuation Guidance Support for Disaster Risk Reduction, in: 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), 2019, pp. 1–6. doi:10.1109/BIGCOMP.2019.8679474.

[14] M. Langerwisch, T. Wittmann, S. Thamke, T. Remmersmann, A. Tiderko, B. Wagner, Heterogeneous teams of unmanned ground and aerial robots for reconnaissance and surveillance - A field experiment, in: 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2013, 2013, pp. 1–6. doi:10.1109/SSRR.2013.6719320.

[15] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, Z. Ming, A Hybrid Path Planning Method in Unmanned Air/Ground Vehicle (UAV/UGV) Cooperative Systems, IEEE Transactions on Vehicular Technology 65 (12) (2016) 9585–9596. doi:10.1109/TVT.2016.2623666.

[16] Q. Han, D. Lu, K. Zhang, X. Du, M. Guizani, A Prediction Method for Destination Based on the Semantic Transfer Model, IEEE Access 7 (2019) 73756–73763. doi:10.1109/access.2019.2918594.

[17] P. C. Besse, B. Guillouet, J. M. Loubes, F. Royer, Destination prediction by trajectory Distribution-Based model, IEEE Transactions on Intelligent Transportation Systems 19 (8) (2018) 2470–2481. arXiv:arXiv:1605.03027v1, doi:10.1109/TITS.2017.2749413.

[18] J. Zhao, J. Xu, R. Zhou, P. Zhao, C. Liu, F. Zhu, On prediction of user destination by sub-trajectory understanding: A Deep Learning based Approach, in: International Conference on Information and Knowledge Management, Proceedings, 2018, pp. 1413–1422. doi:10.1145/3269206.3271708.

[19] M. Zhang, Y. Yang, L. Huang, X. Zhang, Research on destination prediction for urban taxi based on GPS trajectory, International Journal of Performability Engineering 13 (4) (2017) 530–539. doi:10.23940/ijpe.17.04.p20.530539.

[20] X. Li, M. Li, Y. J. Gong, X. L. Zhang, J. Yin, T-DesP: Destination prediction based on big trajectory data, IEEE Transactions on Intelligent Transportation Systems 17 (8) (2016) 2344–2354. doi:10.1109/TITS.2016.2518685.

[21] M. Mukherjee, L. Shu, D. Wang, Survey of fog computing: Fundamental, network applications, and research challenges, IEEE Communications Surveys and Tutorials 20 (3) (2018) 1826–1857. doi:10.1109/COMST.2018.2814571.

[22] R. Mahmud, R. Kotagiri, R. Buyya, Fog Computing: A Taxonomy, Survey and Future Directions, Springer Singapore, Singapore, 2018, pp. 103–130. doi:10.1007/978-981-10-5861-5_5.

[23] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, P. A. Polakos, A comprehensive survey on fog computing: State-of-the-art and research challenges, IEEE Communications Surveys Tutorials 20 (1) (2018) 416–464. doi:10.1109/COMST.2017.2771153.

[24] P. Hu, S. Dhelim, H. Ning, T. Qiu, Survey on fog computing: architecture, key technologies, applications and open issues, Journal of Network and Computer Applications 98 (2017) 27 – 42. doi:https://doi.org/10.1016/j.jnca.2017.09.002.

[25] X. Li, Y. Liu, H. Ji, H. Zhang, V. C. M. Leung, Optimizing resources allocation for fog computing-based internet of things networks, IEEE Access 7 (2019) 64907–64922. doi:10.1109/ACCESS.2019.2917557.

[26] G. Zhang, F. Shen, Z. Liu, Y. Yang, K. Wang, M. T. Zhou, FEMTO: Fair and energy-minimized task offloading for fog-enabled IoT networks, IEEE Internet of Things Journal 6 (3) (2019) 4388–4400. doi:10.1109/JIOT.2018.2887229.

[27] Z. Wei, H. Jiang, Optimal Offloading in Fog Computing Systems With Non-Orthogonal Multiple Access, IEEE Access 6 (2018) 49767–49778. doi:10.1109/ACCESS.2018.2868894.

[28] K. Krishnan, Data-Driven Architecture for Big Data, in: Data Warehousing in the Age of Big Data, Morgan Kaufmann, 2013, pp. 219–240. doi:10.1016/b978-0-12-405891-0.00011-8.

[29] M. L. Wijerathne, L. A. Melgar, M. Hori, T. Ichimura, S. Tanaka, HPC enhanced large urban area evacuation simulations with vision based autonomously navigating multi agents, in: Procedia Computer Science, Vol. 18, Elsevier B.V., 2013, pp. 1515–1524. doi:10.1016/j.procs.2013.05.319.

[30] S. Pluchino, C. M. G. Tribulato, A. Caverzan, A. Mc Quillan, G. P. Cimellaro, S. Mahin, Agent-Based Model for Pedestrians' Evacuation after a Blast Integrated with a Human Behavior Model, in: Structures Congress 2015, American Society of Civil Engineers, 2015, pp. 1506–1517. doi:10.1061/9780784479117.129.

[31] S. Taga, T. Matsuzawa, M. Takimoto, Y. Kambayashi, Multi-Agent Base Evacuation Support System Using MANET, Vietnam Journal of Computer Science 06 (02) (2019) 177–191.

[32] A. Fujihara, H. Miwa, Disaster evacuation guidance using opportunistic communication: The potential for opportunity-based service, Studies in Computational Intelligence 546 (2014) 425–446. doi:10.1007/978-3-319-05029-4_18.

[33] S. D. Ramchurn, T. D. Huynh, F. Wu, Y. Ikuno, J. Flann, L. Moreau, J. E. Fischer, W. Jiang, T. Rodden, E. Simpson, S. Reece, S. Roberts, N. R. Jennings, A disaster response system based on human-agent collectives, Journal of Artificial Intelligence Research 57 (2016) 661–708. doi:10.1613/jair.5098.

[34] D. Liu, Z. Tong, M. Tong, S. Tang, Boundary exploration algorithm of disaster environment of coal mine based on multi-UAVs, in: 2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC), 2017, pp. 240–243. doi:10.1109/ICEIEC.2017.8076553.

[35] S. George, W. Zhou, H. Chenji, M. Won, Y. Lee, A. Pazarloglou, R. Stoleru, P. Barooah, DistressNet: A wireless ad hoc and sensor network architecture for situation management in disaster re-

sponse, IEEE Communications Magazine 48 (3) (2010) 128–136. doi:10.1109/MCOM.2010.5434384.

[36] N. Pogkas, G. E. Karastergios, C. P. Antonopoulos, S. Koubias, G. Papadopoulos, Architecture Design and Implementation of an Ad-Hoc Network for Disaster Relief Operations, IEEE Transactions on Industrial Informatics 3 (1) (2007) 63–72. doi:10.1109/TII.2007.891307.

[37] A. Rego, L. Garcia, S. Sendra, J. Lloret, Software defined networks for traffic management in emergency situations, in: 2018 5th International Conference on Software Defined Systems, SDS 2018, 2018, pp. 45–51. doi:10.1109/SDS.2018.8370421.

[38] K. Goda, N. Mori, T. Yasuda, A. Prasetyo, A. Muhammad, D. Tsujio, Cascading Geological Hazards and Risks of the 2018 Sulawesi Indonesia Earthquake and Sensitivity Analysis of Tsunami Inundation Simulations, Frontiers in Earth Science 7 (oct 2019).

[39] N. C. Bronfman, P. C. Cisternas, P. B. Repetto, J. V. Castañeda, Natural disaster preparedness in a multi-hazard environment: Characterizing the sociodemographic profile of those better (worse) prepared, PLoS ONE 14 (4) (2019) e0214249. doi:10.1371/journal.pone.0214249.

[40] G. Huang, J. He, W. Zhou, G. L. Huang, L. Guo, X. Zhou, F. Tang, Discovery of stop regions for understanding repeat travel behaviors of moving objects, Journal of Computer and System Sciences 82 (4) (2015) 582–593. doi:10.1016/j.jcss.2015.10.006.

[41] Y. Zheng, Trajectory data mining: An overview, ACM Trans. Intell. Syst. Technol. 6 (3) (May 2015). doi:10.1145/2743025.
URL https://doi.org/10.1145/2743025

[42] N. Suri, Z. Zielinski, M. Tortonesi, C. Fuchs, M. Pradhan, K. Wrona, J. Furtak, D. B. Vasilache, M. Street, V. Pellegrini, G. Benincasa, A. Morelli, C. Stefanelli, E. Casini, M. Dyk, Exploiting smart city IoT for disaster recovery operations, in: IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings, Vol. 2018-Janua, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 458–463. doi:10.1109/WF-IoT.2018.8355117.

[43] K. Khosravi, E. Nohani, E. Maroufinia, H. R. Pourghasemi, A GIS-based flood susceptibility assessment and its mapping in Iran: a comparison between frequency ratio and weights-of-evidence bivariate statistical models with multi-criteria decision-making technique, Natural Hazards 83 (2) (2016) 947–987. doi:10.1007/s11069-016-2357-2.

[44] I. Frigerio, S. Ventura, D. Strigaro, M. Mattavelli, M. De Amicis, S. Mugnano, M. Boffi, A GIS-based approach to identify the spatial variability of social vulnerability to seismic hazard in Italy (2016). doi:10.1016/j.apgeog.2016.06.014.

[45] X. Ran, H. Chen, X. Zhu, Z. Liu, J. Chen, DeepDecision: A Mobile Deep Learning Framework for Edge Video Analytics, in: Proceedings - IEEE INFOCOM, Vol. 2018-April, 2018, pp. 1421–1429. doi:10.1109/INFOCOM.2018.8485905.

[46] R. Muñoz, R. Vilalta, N. Yoshikane, R. Casellas, R. Martínez, T. Tsuritani, I. Morita, Integration of IoT, Transport SDN, and Edge/Cloud Computing for Dynamic Distribution of IoT Analytics and Efficient Use of Network Resources, Journal of Lightwave Technology 36 (7) (2018) 1420–1428.

[47] M. Alwateer, S. W. Loke, W. Rahayu, Drone services: An investigation via prototyping and simulation, in: IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings, Vol. 2018-Janua, 2018, pp. 367–370. doi:10.1109/WF-IoT.2018.8355153.

[48] N. Shou, H. Peng, H. Wang, L.-M. Meng, K.-L. Du, An ROIs based pedestrian detection system for single images, in: 2012 5th International Congress on Image and Signal Processing, IEEE, 2012, pp. 1205–1208. doi:10.1109/CISP.2012.6469927.

[49] M. Bertozzi, A. Broggi, A. Lasagni, M. Rose, Infrared stereo vision-based pedestrian detection, in: IEEE Proceedings. Intelligent Vehicles Symposium, 2005., IEEE, 2005, pp. 24–29. doi:10.1109/IVS.2005.1505072.

[50] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, M. Yunsheng, S. Chen, P. Hou, A new deep learning-based food recognition system for dietary

assessment on an edge computing service infrastructure, IEEE Transactions on Services Computing 11 (2) (2017) 249–261.

[51] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding (2014). arXiv:1408.5093.

[52] Caffe2 [online] (2020) [cited 2020-04-23].

[53] C.-C. Tu, P.-Y. Chen, N. Wang, Improving Prediction Efficacy Through Abnormality Detection and Data Preprocessing, IEEE Access 7 (2019) 103794–103805. doi:10.1109/access.2019.2930257.

[54] S. Ramírez-Gallego, S. García, H. Mouriño-Talín, D. Martínez-Rego, V. Bolón-Canedo, A. Alonso-Betanzos, J. M. Benítez, F. Herrera, Data discretization: Taxonomy and big data challenge (jan 2016). doi:10.1002/widm.1173.

[55] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, F. Herrera, A survey on data preprocessing for data stream mining: Current status and future directions, Neurocomputing 239 (2017) 39–57. doi:10.1016/j.neucom.2017.01.078.

[56] P. T. Eugster, P. A. Felber, R. Guerraoui, A.-M. Kermarrec, The many faces of publish/subscribe, ACM Computing Surveys 35 (2) (2003) 114–131.

[57] M. Tortonesi, J. Michaelis, A. Morelli, N. Suri, M. A. Baker, SPF: An SDN-based middleware solution to mitigate the IoT information explosion, in: 2016 IEEE Symposium on Computers and Communication (ISCC), IEEE, 2016, pp. 435–442.
URL http://ieeexplore.ieee.org/document/7543778/

[58] J. McClurg, H. Hojjat, N. Foster, P. Cěrnyá, Event-driven network programming, in: Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), Vol. 13-17-June, Association for Computing Machinery, New York, New York, USA, 2016, pp. 369–385. doi:10.1145/2908080.2908097.

[59] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, D. Walker, Frenetic, in: Proceeding of the 16th ACM SIGPLAN international conference on Functional programming - ICFP '11, ACM Press, New York, New York, USA, 2011, p. 279. doi:10.1145/2034773.2034812.

[60] A. Sapio, I. Abdelaziz, A. Aldilaijan, M. Canini, P. Kalnis, In-network computation is a dumb idea whose time has come, in: HotNets 2017 - Proceedings of the 16th ACM Workshop on Hot Topics in Networks, Association for Computing Machinery, Inc, New York, NY, USA, 2017, pp. 150–156. doi:10.1145/3152434.3152461.

[61] C. Monsanto, J. Reich, N. Foster, J. Rexford, D. Walker, Composing Software-defined Networks, in: 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13), 2013, pp. 1–13.

[62] C. J. Anderson, N. Foster, A. Guha, J. B. Jeannin, D. Kozen, C. Schlesinger, D. Walker, NetKAT: Semantic foundations for networks, in: ACM SIGPLAN Notices, Vol. 49, 2014, pp. 113–126. doi:10.1145/2578855.2535862.

[63] K. E. Benson, G. Wang, N. Venkatasubramanian, Y. J. Kim, Ride: A resilient IoT data exchange middleware leveraging SDN and edge cloud resources, in: Proceedings - ACM/IEEE International Conference on Internet of Things Design and Implementation, IoTDI 2018, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 72–83.

[64] M. A. Tariq, B. Koldehofe, S. Bhowmik, K. Rothermel, PLEROMA: A SDN-based high performance publish/subscribe middleware, in: Proceedings of the 15th International Middleware Conference, Middleware 2014, Association for Computing Machinery, Inc, 2014, pp. 217–228.

[65] A. Hakiri, P. Berthou, A. Gokhale, S. Abdellatif, Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications, IEEE Communications Magazine 53 (9) (2015) 48–54.

[66] L. Scalzotto, K. E. Benson, G. Bouloukakis, P. Bellavista, V. Issarny, S. Mehrotra, N. Venkatasubramanian, An implementation experience with SDN-enabled iot data exchange middleware, in: Middleware 2018

- Proceedings of the 2018 ACM/IFIP/USENIX Middleware Conference (Posters), Association for Computing Machinery, Inc, 2018, pp. 21–22.

[67] Y. Kang, I. Park, J. Rhee, Y. Lee, MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data, IEEE Sensors Journal 16 (2) (2016) 485–497.

[68] S. Qian, G. Wu, J. Huang, T. Das, Benchmarking modern distributed streaming platforms, in: Proceedings of the IEEE International Conference on Industrial Technology, Vol. 2016-May, Institute of Electrical and Electronics Engineers Inc., 2016, pp. 592–598.

[69] Apache Cassandra [online] (2020) [cited 2020-04-23].

[70] Apache HBase – Apache HBase™ Home [online] (2020) [cited 2020-04-23].

[71] Apache Spark™ - Unified Analytics Engine for Big Data [online] (2020) [cited 2020-04-23].

[72] Apache Samza [online] (2020) [cited 2020-04-23].

[73] Apache Kafka [online] (2020) [cited 2020-04-23].

[74] Apache Storm [online] (2020) [cited 2020-04-23].

[75] Y. H. Chen, C. C. Hu, E. H. K. Wu, S. M. Chuang, G. H. Chen, A Delay-Sensitive Multicast Protocol for Network Capacity Enhancement in Multirate MANETs, IEEE Systems Journal 12 (1) (2018) 926–937. doi:10.1109/JSYST.2017.2677952.

[76] L. Qiu, G. Cao, Popularity-Aware Caching Increases the Capacity of Wireless Networks, IEEE Transactions on Mobile Computing 19 (1) (2020) 173–187. doi:10.1109/TMC.2019.2892419.

[77] M. A. Salem, I. F. Tarrad, M. I. Youssef, S. M. A. El-Kader, QoS categories activeness-aware adaptive EDCA algorithm for dense IoT networks, International Journal of Computer Networks and Communications 11 (3) (2019) 67–83. doi:10.5121/ijcnc.2019.11305.

[78] D. I. George Amalarethinam, P. Mercy, An analysis on quality of service (Qos) based routing in internet of things (IOT), International Journal of Advanced Science and Technology 29 (5 Special Issue) (2020) 488–496.

[79] F. S. Abkenar, A. Jamalipour, EBA: Energy Balancing Algorithm for Fog-IoT Networks, IEEE Internet of Things Journal 6 (4) (2019) 6843–6849. doi:10.1109/JIOT.2019.2911953.

[80] H. Al-Hamadi, M. Saoud, I.-R. Chen, J.-H. Cho, Optimizing the Lifetime of IoT-Based Star and Mesh Networks, IEEE Access 8 (2020) 63090–63105. doi:10.1109/ACCESS.2020.2983873.

[81] S. Dawaliby, A. Bradai, Y. Pousset, Adaptive dynamic network slicing in LoRa networks, Future Generation Computer Systems 98 (2019) 697–707. doi:10.1016/j.future.2019.01.042.

[82] T. Zhang, J. Jin, X. Zheng, Y. Yang, Rate-Adaptive Fog Service Platform for Heterogeneous IoT Applications, IEEE Internet of Things Journal 7 (1) (2020) 176–188. doi:10.1109/JIOT.2019.2945328.

[83] K. Cao, G. Xu, J. Zhou, T. Wei, M. Chen, S. Hu, QoS-Adaptive Approximate Real-Time Computation for Mobility-Aware IoT Lifetime Optimization, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 38 (10) (2019) 1799–1810. doi:10.1109/TCAD.2018.2873239.

[84] L. Song, K. K. Chai, Y. Chen, J. Schormans, J. Loo, A. Vinel, QoS-Aware Energy-Efficient Cooperative Scheme for Cluster-Based IoT Systems, IEEE Systems Journal 11 (3) (2017) 1447–1455. doi:10.1109/JSYST.2015.2465292.

[85] S. Muralidharan, B. J. R. Sahu, N. Saxena, A. Roy, PPT: A Push Pull Traffic Algorithm to Improve QoS Provisioning in IoT-NDN Environment, IEEE Communications Letters 21 (6) (2017) 1417–1420. doi:10.1109/LCOMM.2017.2677922.

[86] M. Li, Z. Zhang, K. Huang, T. Tan, Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection, in: 2008 19th International Conference on Pattern Recognition, 2008, pp. 1–4. doi:10.1109/ICPR.2008.4761705.

[87] K. Chen, C. C. Loy, S. Gong, T. Xiang, Feature mining for localised crowd counting, in: BMVC 2012 - Electronic Proceedings of the British Machine Vision Conference 2012, 2012, pp. 21.1–21.11. doi:10.5244/C.26.21.

[88] A. B. Chan, N. Vasconcelos, Bayesian poisson regression for crowd counting, in: Proceedings of the IEEE International Conference on Computer Vision, 2009, pp. 545–551. doi:10.1109/ICCV.2009.5459191.

[89] D. Ryan, S. Denman, C. Fookes, S. Sridharan, Crowd counting using multiple local features, in: DICTA 2009 - Digital Image Computing: Techniques and Applications, 2009, pp. 81–88. doi:10.1109/DICTA.2009.22.

[90] V. Q. Pham, T. Kozakaya, O. Yamaguchi, R. Okada, COUNT forest: Co-voting uncertain number of targets using random forest for crowd density estimation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3253–3261. doi:10.1109/ICCV.2015.372.

[91] V. Lempitsky, A. Zisserman, Learning to count objects in images, in: Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010, 2010, pp. 1324–1332.

[92] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444. doi:10.1038/nature14539.
URL https://doi.org/10.1038/nature14539

[93] C. Wang, H. Zhang, L. Yang, S. Liu, X. Cao, Deep people counting in extremely dense crowds, in: Proceedings of the 23rd ACM International Conference on Multimedia, MM '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 1299–1302. doi:10.1145/2733373.2806337.

[94] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, W. Xu, CNN-RNN: A Unified Framework for Multi-label Image Classification, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 2285–2294. arXiv:1604.04573, doi:10.1109/CVPR.2016.251.

[95] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708. doi:10.1109/CVPR.2014.220.

[96] B. Suter, T. V. Lakshman, D. Stiliadis, A. K. Choudhury, Design considerations for supporting TCP with per-flow queueing, in: Proceedings - IEEE INFOCOM, 1998, pp. 299–306. doi:10.1109/infcom.1998.659666.

[97] F. M. Anjum, L. Tassiulas, Fair bandwidth sharing among adaptive and non-adaptive flows in the Internet, in: Proceedings - IEEE INFOCOM, Vol. 3, 1999, pp. 1412–1420. doi:10.1109/INFCOM.1999.752161.

[98] D. Lin, R. Morris, Dynamics of random early detection, Computer Communication Review 27 (4) (1997) 127–136. doi:10.1145/263109.263154.

[99] T. J. Ott, T. V. Lakshman, L. H. Wong, SRED: Stabilized RED, in: Proceedings - IEEE INFOCOM, 1999, pp. 1346–1355. doi:10.1109/INFCOM.1999.752153.

[100] Rong Pan, B. Prabhakar, K. Psounis, CHOKe - a stateless active queue management scheme for approximating fair bandwidth allocation, in: Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), Vol. 2, IEEE, 2000, pp. 942–951. doi:10.1109/INFCOM.2000.832269.

[101] W. Feng, D. D. Kandlur, D. Saha, K. G. Shin, Stochastic fair blue: A queue management algorithm for enforcing fairness, in: Proceedings - IEEE INFOCOM, Vol. 3, 2001, pp. 1520–1529. doi:10.1109/infcom.2001.916648.

[102] G. Chatranon, M. A. Labrador, S. Banerjee, Black: Detection and preferential dropping of high bandwidth unresponsive flows, in: IEEE International Conference on Communications, 2003, pp. 664–668. doi:10.1109/icc.2003.1204258.

[103] M. K. Chan, M. Hamdi, An active queue management scheme based on a capture-recapture model, IEEE Journal on Selected Areas in Communications 21 (4) (2003) 572–583. doi:10.1109/JSAC.2003.810499.

[104] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, O. M. Caicedo, A comprehensive survey on machine learning for networking: evolution, applications and research opportunities, Journal of Internet Services and Applications 9 (1) (2018) 1–99. doi:10.1186/s13174-018-0087-2.

[105] M. Wang, Y. Cui, X. Wang, S. Xiao, J. Jiang, Machine Learning for Networking: Workflow, Advances and Opportunities, IEEE Network 32 (2) (2018) 92–99. arXiv:1709.08339, doi:10.1109/MNET.2017.1700200.

[106] S. Chen, H. Wang, F. Xu, Y.-Q. Jin, Target classification using the deep convolutional networks for sar images, IEEE Transactions on Geoscience and Remote Sensing (2016). doi:10.1109/TGRS.2016.2551720.

[107] L. Paradis, Q. Han, A survey of fault management in wireless sensor networks, Journal of Network and Systems Management 15 (2) (2007) 171–190. doi:10.1007/s10922-007-9062-0.

[108] S. Liu, Q. Qu, S. Wang, Rationality Analytics from Trajectories, ACM Trans. Knowl. Discov. Data 10 (1) (jul 2015). doi:10.1145/2735634. URL https://doi.org/10.1145/2735634

[109] K. Kucuk, C. Bayilmis, A. F. Sonmez, S. Kacar, Crowd sensing aware disaster framework design with IoT technologies, Journal of Ambient Intelligence and Humanized Computing 11 (4) (2020) 1709–1725. doi:10.1007/s12652-019-01384-1.

[110] I. F. Akyildiz, W.-Y. Lee, K. R. Chowdhury, CRAHNs: Cognitive radio ad hoc networks, Ad Hoc Networks 7 (5) (2009) 810–836. doi:10.1016/j.adhoc.2009.01.001.

[111] Z. Guan, T. Ji, X. Qian, Y. Ma, X. Hong, A Survey on Big Data Pre-processing, in: 2017 5th Intl Conf on Applied Computing and Information Technology/4th Intl Conf on Computational Science/Intelligence

and Applied Informatics/2nd Intl Conf on Big Data, Cloud Computing, Data Science (ACIT-CSII-BCD), IEEE, 2017, pp. 241–247. doi:10.1109/ACIT-CSII-BCD.2017.49.

[112] V. A. Sindagi, V. M. Patel, A survey of recent advances in CNN-based single image crowd counting and density estimation, Pattern Recognition Letters 107 (2018) 3–16. arXiv:1707.01202, doi:10.1016/j.patrec.2017.07.007.

[113] J. Chen, X. Ran, Deep Learning With Edge Computing: A Review, Proceedings of the IEEE 107 (8) (2019) 1655–1674. doi:10.1109/JPROC.2019.2921977.

[114] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, L. Tang, Neurosurgeon: Collaborative intelligence between the cloud and mobile edge, in: International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS, Vol. Part F127193, Association for Computing Machinery, 2017, pp. 615–629. doi:10.1145/3037697.3037698.

[115] J. Mao, X. Chen, K. W. Nixon, C. Krieger, Y. Chen, MoDNN: Local distributed mobile computing system for Deep Neural Network, in: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, IEEE, 2017, pp. 1396–1401. doi:10.23919/DATE.2017.7927211.

[116] Z. Zhao, K. M. Barijough, A. Gerstlauer, DeepThings: Distributed adaptive deep learning inference on resource-constrained IoT edge clusters, in: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 37, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 2348–2359. doi:10.1109/TCAD.2018.2858384.

[117] T. Y.-H. Chen, L. Ravindranath, S. Deng, P. Bahl, H. Balakrishnan, Glimpse: Continuous, real-time object recognition on mobile devices, in: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, 2015, pp. 155–168.

[118] Tensorflow Lite [online] (2020) [cited 2020-06-02].

[119] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, S. Cla, Middleware for internet of things: A survey, IEEE Internet of Things Journal 3 (1) (2016) 70–95.

[120] E. Hwang, K. J. Kim, A. Lyakhov, B. D. Choi, Performance analysis of bandwidth requests under unicast, multicast and broadcast pollings in IEEE 802.16d/e, Telecommunication Systems 50 (1) (2012) 15–30. doi:10.1007/s11235-010-9386-z.

[121] A. Samuvlov, D. Moltchanov, A. Krupko, R. Kovalchukov, F. Moskaleva, Y. Gaidamaka, Performance Analysis of Mixture of Unicast and Multicast Sessions in 5G NR Systems, in: International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, IEEE, 2019, pp. 1–7. doi:10.1109/ICUMT.2018.8631230.

[122] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos, A Survey of information-centric networking research, IEEE Communications Surveys and Tutorials 16 (2) (2014) 1024–1049.

[123] M. Amadeo, C. Campolo, A. Molinaro, G. Ruggeri, Content-centric wireless networking: A survey, Computer Networks 72 (2014) 1–13.

[124] M. Amadeo, C. Campolo, A. Molinaro, Information-centric networking for connected vehicles: A survey and future perspectives, IEEE Communications Magazine 54 (2) (2016) 98–104.

[125] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, W. Chou, Research challenges for traffic engineering in software defined networks, IEEE Network 30 (3) (2016) 52–58.

[126] J. Liu, Y. Li, M. Chen, W. Dong, D. Jin, Software-defined internet of things for smart urban sensing, IEEE Communications Magazine 53 (9) (2015) 55–63.

[127] N. B. Truong, G. M. Lee, Y. Ghamri-Doudane, Software defined networking-based vehicular Adhoc Network with Fog Computing, in: Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, no. March 2016, 2015, pp. 1202–1207.

[128] K. Al-Gumaei, K. Schuba, A. Friesen, S. Heymann, C. Pieper, F. Pethig, S. Schriegel, A Survey of Internet of Things and Big Data integrated Solutions for Industrie 4.0, in: IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Vol. 2018-Septe, Institute of Electrical and Electronics Engineers Inc., 2018, pp. 1417–1424. doi:10.1109/ETFA.2018.8502484.

[129] M. Haghi Kashani, A. M. Rahmani, N. Jafari Navimipour, Quality of service-aware approaches in fog computing, International Journal of Communication Systems 33 (8) (2020) e4340. doi:10.1002/dac.4340. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.4340

[130] Z. Chang, Z. Zhou, T. Ristaniemi, Z. Niu, Energy Efficient Optimization for Computation Offloading in Fog Computing System, in: GLOBECOM 2017 - 2017 IEEE Global Communications Conference, IEEE, 2017, pp. 1–6. doi:10.1109/GLOCOM.2017.8254207.

[131] Y. Nan, W. Li, W. Bao, F. C. Delicato, P. F. Pires, A. Y. Zomaya, A dynamic tradeoff data processing framework for delay-sensitive applications in Cloud of Things systems, Journal of Parallel and Distributed Computing 112 (2018) 53–66. doi:10.1016/j.jpdc.2017.09.009.

[132] L. Liu, Z. Chang, X. Guo, S. Mao, T. Ristaniemi, Multiobjective Optimization for Computation Offloading in Fog Computing, IEEE Internet of Things Journal 5 (1) (2018) 283–294. doi:10.1109/JIOT.2017.2780236.

[133] Y. Dong, S. Guo, J. Liu, Y. Yang, Energy-Efficient Fair Cooperation Fog Computing in Mobile Edge Networks for Smart City, IEEE Internet of Things Journal 6 (5) (2019) 7543–7554. doi:10.1109/JIOT.2019.2901532.

[134] H. Li, K. Ota, M. Dong, Deep reinforcement scheduling for mobile crowdsensing in fog computing, ACM Transactions on Internet Technology 19 (2) (2019). doi:10.1145/3234463.

[135] L. Ni, J. Zhang, C. Jiang, C. Yan, K. Yu, Resource Allocation Strategy in Fog Computing Based on Priced Timed Petri Nets, IEEE Internet

of Things Journal 4 (5) (2017) 1216–1228. doi:10.1109/JIOT.2017.2709814.

[136] L. Li, Q. Guan, L. Jin, M. Guo, Resource Allocation and Task Offloading for Heterogeneous Real-Time Tasks With Uncertain Duration Time in a Fog Queueing System, IEEE Access 7 (2019) 9912–9925. doi:10.1109/ACCESS.2019.2891130.

[137] A. Yousefpour, A. Patil, G. Ishigaki, I. Kim, X. Wang, H. C. Cankaya, Q. Zhang, W. Xie, J. P. Jue, FOGPLAN: A lightweight QoS-aware dynamic fog service provisioning framework, IEEE Internet of Things Journal 6 (3) (2019) 5080–5096. doi:10.1109/JIOT.2019.2896311.

[138] J. Yao, N. Ansari, Fog Resource Provisioning in Reliability-Aware IoT Networks, IEEE Internet of Things Journal 6 (5) (2019) 8262–8269. doi:10.1109/JIOT.2019.2922585.

[139] A. Brogi, S. Forti, QoS-aware deployment of IoT applications through the fog, IEEE Internet of Things Journal 4 (5) (2017) 1–8. doi:10.1109/JIOT.2017.2701408.

[140] A. Abdelghany, K. Abdelghany, H. Mahmassani, W. Alhalabi, Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities, European Journal of Operational Research (2014). doi:10.1016/j.ejor.2014.02.054.

[141] J. E. Almeida, R. J. Rossetti, F. Aguiar, E. Oliveira, Crowd Simulation Applied to Emergency and Evacuation Scenarios, in: Advances in Artificial Transportation Systems and Simulation, Academic Press, 2015, pp. 149–161. doi:10.1016/B978-0-12-397041-1.00008-X.

[142] W. K. Chow, C. M. Ng, Waiting time in emergency evacuation of crowded public transport terminals, Safety Science (2008). doi:10.1016/j.ssci.2007.01.015.

[143] J. D. Sime, Crowd psychology and engineering, Safety Science (1995). doi:10.1016/0925-7535(96)81011-3.

[144] M. Enzweiler, D. M. Gavrila, Monocular pedestrian detection: Survey and experiments, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, pp. 2179–2195. doi:10.1109/TPAMI.2008.260.

[145] O. Tuzel, F. Porikli, P. Meer, Pedestrian detection via classification on Riemannian manifolds, IEEE Transactions on Pattern Analysis and Machine Intelligence (2008). doi:10.1109/TPAMI.2008.75.

[146] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, 2005, pp. 886–893 vol. 1. doi:10.1109/CVPR.2005.177.

[147] M. Fu, P. Xu, X. Li, Q. Liu, M. Ye, C. Zhu, Fast crowd density estimation with convolutional neural networks, Engineering Applications of Artificial Intelligence 43 (2015) 81 – 88. doi:https://doi.org/10.1016/j.engappai.2015.04.006.

[148] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. Lecun, Pedestrian detection with unsupervised multi-stage feature learning, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3626–3633. doi:10.1109/CVPR.2013.465.

[149] L. Boominathan, S. S. S. Kruthiventi, R. V. Babu, Crowdnet: A deep convolutional network for dense crowd counting, in: Proceedings of the 24th ACM International Conference on Multimedia, MM '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 640–644. doi:10.1145/2964284.2967300.
URL https://doi.org/10.1145/2964284.2967300

[150] Y. Zhang, D. Zhou, S. Chen, S. Gao, Y. Ma, Single-image crowd counting via multi-column convolutional neural network, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 589–597. doi:10.1109/CVPR.2016.70.

[151] C. Shang, H. Ai, B. Bai, End-to-end crowd counting via joint learning local and global count, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 1215–1219. doi:10.1109/ICIP.2016.7532551.

[152] B. Sheng, C. Shen, G. Lin, J. Li, W. Yang, C. Sun, Crowd counting via weighted vlad on a dense attribute feature map, IEEE Transactions on Circuits and Systems for Video Technology 28 (8) (2018) 1788–1797. doi:10.1109/TCSVT.2016.2637379.

[153] V. A. Sindagi, V. M. Patel, Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting, in: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2017, pp. 1–6. doi:10.1109/AVSS.2017.8078491.

[154] P. Zhang, H. Yue, X. Zhang, C. Shao, W. Gao, Modeling the Equilibrium Road Network Capacity, IEEE Access 7 (2019) 168029–168047. doi:10.1109/ACCESS.2019.2946977.

[155] T. Anwar, C. Liu, H. L. Vu, C. Leckie, Partitioning road networks using density peak graphs: Efficiency vs. accuracy, Information Systems (2017). doi:10.1016/j.is.2016.09.006.

[156] F. Su, H. Dong, L. Jia, Z. Tian, X. Sun, Space-time correlation analysis of traffic flow on road network, International Journal of Modern Physics B (2017). doi:10.1142/S0217979217500278.

[157] P. Lin, J. Weng, B. Yin, X. Zhou, Urban Road Network Operation Quality Evaluation Method Based on High-Frequency Trajectory Data, in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 2018, pp. 3602–3607. doi:10.1109/ITSC.2018.8569750.

[158] M. Xu, J. Wu, M. Liu, Y. Xiao, H. Wang, D. Hu, Discovery of Critical Nodes in Road Networks Through Mining from Vehicle Trajectories, IEEE Transactions on Intelligent Transportation Systems (2019). doi:10.1109/TITS.2018.2817282.

[159] L. Wu, Z. Huang, J. Wu, Z. Gao, D. Qin, A Day-to-Day Stochastic Traffic Flow Assignment Model Based on Mixed Regulation, IEEE Access (2020). doi:10.1109/ACCESS.2019.2962864.

[160] M. Bani Younes, A. Boukerche, Intelligent Traffic Light Controlling Algorithms Using Vehicular Networks, IEEE Transactions on Vehicular Technology (2016). doi:10.1109/TVT.2015.2472367.

[161] Y. Xu, H. Chen, Q.-J. Kong, X. Zhai, Y. Liu, Urban traffic flow prediction: A spatio-temporal variable selection-based approach, Journal of Advanced Transportation (2016). doi:10.1002/atr.1356.

[162] F. Crawford, D. P. Watling, R. D. Connors, A statistical method for estimating predictable differences between daily traffic flow profiles, Transportation Research Part B: Methodological (2017). doi:10.1016/j.trb.2016.11.004.

[163] Y. Zhang, T. Cheng, Y. Ren, A graph deep learning method for short-term traffic forecasting on large road networks, Computer-Aided Civil and Infrastructure Engineering (2019). doi:10.1111/mice.12450.

[164] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, 2018.

[165] W. Wisetjindawat, A. Kermanshah, S. Derrible, M. Fujita, Stochastic modeling of road system performance during multihazard events: Flash floods and earthquakes, Journal of Infrastructure Systems 23 (4) (dec 2017). doi:10.1061/(ASCE)IS.1943-555X.0000391.

[166] S. S. Sarma, K. Sinha, G. Chakraborty, B. P. Sinha, Distributed algorithm for traffic dissemination in manhattan networks with optimal routing-time, in: Proceedings of the ACM Symposium on Applied Computing, 2017, pp. 499–505. doi:10.1145/3019612.3019702.

[167] B. U. Braden, D. M. L. Clark, J. U. Crowcroft, B. C. S. Davie, S. C. S. Deering, D. U. Estrin, S. L. Floyd, V. L. Jacobson, G. F. Minshall, C. B. Partridge, L. U. o. A. Peterson, K. A. L. R. Ramakrishnan, S. X. P. Shenker, J. M. L. Wroclawski, L. U. Zhang, RFC 2309: Recommendations on Queue Management and Congestion Avoidance in the Internet, Tech. rep., IETF (1998).
URL https://tools.ietf.org/html/rfc2309

[168] R. Adams, Active queue management: A survey, IEEE Communications Surveys and Tutorials (2013). doi:10.1109/SURV.2012.082212.00018.

[169] S. Floyd, V. Jacobson, Random Early Detection Gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking (1993). doi:10.1109/90.251892.

[170] G. Chatranon, M. A. Labrador, S. Banerjee, A survey of TCP-friendly router-based AQM schemes (sep 2004). doi:10.1016/j.comcom.2004.05.001.

[171] F. Li, J. Sun, M. Zukerman, Z. Liu, Q. Xu, S. Chan, G. Chen, K.-T. Ko, A comparative simulation study of TCP/AQM systems for evaluating the potential of neuron-based AQM schemes, Journal of Network and Computer Applications 41 (2014) 274–299. doi:10.1016/j.jnca.2014.01.005.

[172] B. Hariri, N. Sadati, NN-RED: an AQM mechanism based on neural networks, Electronics Letters 43 (19) (2007) 1053. doi:10.1049/el:20071791.

[173] J. Sun, S. Chan, K.-t. Ko, G. Chen, M. Zukerman, Neuron PID: A Robust AQM Scheme, in: 2006 The Australian Telecommunication Networks & Applications Conference - ATNAC, University of Melbourne, 2006, pp. 259–262.

[174] C. Zhou, D. Di, Q. Chen, J. Guo, An adaptive AQM algorithm based on neuron reinforcement learning, in: 2009 IEEE International Conference on Control and Automation, IEEE, 2009, pp. 1342–1346. doi:10.1109/ICCA.2009.5410198.

[175] J. Sun, M. Zukerman, An adaptive neuron AQM for a stable internet, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 4479 LNCS, Springer, Berlin, Heidelberg, 2007, pp. 844–854. doi:10.1007/978-3-540-72606-7_72.

[176] L. Chen, M. Lv, G. Chen, A system for destination and future route prediction based on trajectory mining, in: Pervasive and Mobile Computing, Vol. 6, 2010, pp. 657–676. doi:10.1016/j.pmcj.2010.08.004.

[177] A. Y. Xue, J. Qi, X. Xie, R. Zhang, J. Huang, Y. Li, Solving the Data Sparsity Problem in Destination Prediction, The VLDB Journal 24 (2) (2015) 219–243. doi:10.1007/s00778-014-0369-7.
URL https://doi.org/10.1007/s00778-014-0369-7

[178] P. Afsar, P. Cortez, H. Santos, Automatic human trajectory destination prediction from video, Expert Systems with Applications 110 (2018) 41–51. doi:10.1016/j.eswa.2018.03.035.

[179] J. J.-C. Ying, W.-C. Lee, V. S. Tseng, Mining Geographic-Temporal-Semantic Patterns in Trajectories for Location Prediction, ACM Trans. Intell. Syst. Technol. 5 (1) (jan 2014). doi:10.1145/2542182.2542184. URL https://doi.org/10.1145/2542182.2542184

[180] W. X. Zhao, N. Zhou, W. Zhang, J.-R. Wen, S. Wang, E. Y. Chang, A Probabilistic Lifestyle-Based Trajectory Model for Social Strength Inference from Human Trajectory Data, ACM Trans. Inf. Syst. 35 (1) (sep 2016). doi:10.1145/2948064.
URL https://doi.org/10.1145/2948064

[181] C.-C. Hung, W.-C. Peng, W.-C. Lee, Clustering and Aggregating Clues of Trajectories for Mining Trajectory Patterns and Routes, The VLDB Journal 24 (2) (2015) 169–192. doi:10.1007/s00778-011-0262-6.

[182] RabbitMQ [online] (2020) [cited 2020-06-02].

[183] P. Dobbelaere, K. S. Esmaili, Industry paper: Kafka versus RabbitMQ: A comparative study of two industry reference publish/subscribe implementations, in: DEBS 2017 - Proceedings of the 11th ACM International Conference on Distributed Event-Based Systems, Association for Computing Machinery, Inc, New York, New York, USA, 2017, pp. 227–238. doi:10.1145/3093742.3093908.

[184] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, H. S. Mamede, Machine learning in software defined networks: Data collection and traffic classification, in: Proceedings - International Conference on Network Protocols, ICNP, IEEE Computer Society, 2016, pp. 1–5. doi:10.1109/ICNP.2016.7785327.

[185] T. Kohler, R. Mayer, F. Dürr, M. Maaß, S. Bhowmik, K. Rothermel, P4CEP: Towards in-network Complex Event Processing, in: NetCompute 2018 - Proceedings of the 2018 Morning Workshop on In-Network Computing, Part of SIGCOMM 2018, Association for Computing Machinery, Inc, 2018, pp. 33–38. arXiv:1806.04385, doi:10.1145/3229591.3229593.

[186] M. T. Arashloo, Y. Koral, M. Greenberg, J. Rexford, D. Walker, SNAP: Stateful network-wide abstractions for packet processing, in: SIGCOMM 2016 - Proceedings of the 2016 ACM Conference on Special Interest Group on Data Communication, Association for Computing Machinery, Inc, New York, New York, USA, 2016, pp. 29–43. arXiv:1512.00822, doi:10.1145/2934872.2934892.

[187] A. Voellmy, J. Wang, Y. R. Yang, B. Ford, P. Hudak, Maple, in: Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM - SIGCOMM '13, ACM Press, New York, New York, USA, 2013, p. 87. doi:10.1145/2486001.2486030.

[188] R. Soulé, S. Basu, P. J. Marandi, F. Pedone, R. Kleinberg, E. G. Sirer, N. Foster, Merlin, in: Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies - CoNEXT '14, ACM Press, New York, New York, USA, 2014, pp. 213–226. arXiv:1407.1199, doi:10.1145/2674005.2674989.

[189] T. Nelson, A. D. Ferguson, M. J. Scheer, S. Krishnamurthi, Tierless programming and reasoning for software-defined networks, in: Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2014, 2014, pp. 519–531.

[190] H. Phan, D. Huynh, Y. He, M. Savvides, Z. Shen, Mobinet: A mobile binary network for image classification (2019). arXiv:1907.12629.

[191] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[192] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[193] M. Verhelst, B. Moons, Embedded Deep Neural Network Processing: Algorithmic and Processor Techniques Bring Deep Learning to IoT and Edge Devices, IEEE Solid-State Circuits Magazine 9 (4) (2017) 55–65. doi:10.1109/MSSC.2017.2745818.