# ADMITS: Architecting Distributed Monitoring and Analytics in IoT-based Disaster Scenarios

**Rafael Pasquini[1], Rodrigo S. Miani[1], Paulo R. S. L. Coelho[1], Augusto V. Neto[2,3]**
**Nicolás Hidalgo[4], Martín Gutiérrez[4], Erika Rosas[5], Javier Baliosian[6], Eduardo Grampín[6]**

[1]Faculdade de Computação (FACOM/UFU)
Caixa Postal 593 – 38.400-902 – Uberlândia – MG – Brazil

[2]Departamento de Informática e Matemática Aplicada (DIMAp/UFRN)
Campus Universitário Lagoa Nova – 59078-970 – Natal – RN – Brazil

[3]Instituto de Telecomunicações – Universidade de Aveiro
Campus Universitário – 3810-193 – Aveiro, Portugal

[4]Escuela de Informática y Telecomunicaciones (EIT/UDP)
Vergara 432 - Santiago, Chile

[5]Departamento de Informática, Universidad Técnica Federico Santa María
Vicuña Mackenna 3939, Santiago, Chile

[6]Facultad de Ingeniería, Universidad de la República, Uruguay
Julio HErrera y Reissig 565, Montevideo, Uruguay

{pasquini, miani, paulocoelho}@ufu.br, augusto@dimap.ufrn.br,

{nicolas.hidalgoc, martin.gutierrez}@mail.udp.cl, erosas@inf.utfsm.cl

{baliosian, grampin}@fing.edu.uy

***Abstract.*** *The ADMITS project aims to develop algorithms, protocols and architectures to enable a distributed computing environment to provide support for monitoring, failure detection, and analytics in IoT disaster scenarios. We face a context where, every year, millions of people are affected by natural and man-made disasters, whereby governments all around the world spend huge amounts of resources on preparation, immediate response, and reconstruction. Recently, the Internet of Things (IoT) paradigm has been extensively used for efficiently managing disaster scenarios, such as volcanic disasters, floods, forest fire, landslides, earthquakes, urban disasters, industrial and terrorists attacks, and so on. However, in a disaster scenario the communication/processing infrastructure and the devices themselves may fail, producing either temporary or permanent network partitions and loss of information. Moreover, it is expected that in the years to come, IoT will generate large amounts of data, making processing and analysis challenging in time-critical applications. Considering such challenges, ADMITS targets the development of a architecture in which IoT, Fog, and Cloud computing technologies participate to provide required capabilities for IoT data analytics, real-time stream processing, and failure monitoring for environments potentially subject to disasters. In this positional paper, we discuss the motivation, objectives, architecture, research challenges (and how to overcome them) and initial efforts for the ADMITS project.*

## 1. Introduction

Every year, millions of people are affected by natural disasters such as earthquakes, tsunamis, volcano eruptions, hurricanes, tornadoes and floods, and governments all around the world spend huge amounts of resources on the reconstruction and the preparation for such calamities [Rosas et al. 2016]. Only in 2016, the number of natural hazards that hit the world was of 342, affecting a number of 564 million people and producing an economic damage of US$154 billion [Guha-Sapir et al. 2016]. Moreover, man-made disasters may lead to huge destruction in cases of terrorist attacks or war related events.

Recently, the Internet of Things (IoT) paradigm has been proposed to manage disaster scenarios. IoT refers to the seamless communication, monitoring, and management of smart embedded devices with its counterpart, i.e. analog objects or things. In the coming years, IoT is expected to bridge diverse technologies to enable new applications by connecting physical objects together in support of intelligent decision making [Al-Fuqaha et al. 2015]. In disasters, IoT provides value to emergency response operations in terms of improving cooperation, forecasting, and situation awareness [Yang et al. 2013]. They have been proposed to localize victims in post disaster environments, achieve situation awareness, and monitor the environment [Ray et al. 2017].

In a disaster scenario the communication/processing infrastructure may fail, producing temporary or permanent network partitions and loss of information. The IoT infrastructure may involve wireless sensor networks communicating with Cloud infrastructure where data is analyzed, for example. Moreover, IoT potentially produces a large amount of data to analyze (It is expected that IoT will generate 4.4 trillion GB by 2020 [Siozios et al. 2018]). In order to cope with large amounts of data, the Kappa architecture is proposed for the processing and visualization of data streams [Kreps 2014]. Although such an architecture can support the scalable processing and the deployment of several data analytic algorithms on wireless sensor networks (WSN) and IoT data, they lack robust and cost-communication effective approaches for the deployment of large scale platforms.

Therefore, other architectures where data is analyzed at the edge of the network have been proposed recently [Chiang and Zhang 2016][Cisco 2015]. In this big data context, it is a great advantage to avoid the movement of a large volume of data towards the Internet core, so that using Fog Computing, which provides computational resources placed in the edge of the network and near to the sensors and IoT devices is an interesting paradigm to study. Traditional data analysis however, has not yet addressed the constraints of such distributed environment: to execute in a distributed fashion over devices with limited energy and computational resources, in a context of possible failure, and real-time requirements that a pre/post disaster scenario imposes. In this scenario, a new generation of distributed analysis methods are required [Stolpe 2016].

The authors in [Ray et al. 2017] provide a good survey of IoT supported protocols for disaster management, IoT cost-effective available market solutions for disaster management and IoT-based applications for disaster management systems. Moreover, IoT has been used to provide services related to these disaster scenarios, such as crowd-sourced IoT framework and real-time stream processing [Rauniyar et al. 2016]. IoT-based disaster management usually includes sensor and/or smart devices networks used to monitor a disaster sensible region (e.g., forest fire, flood, landslide disaster management, etc.). On

the other hand, devices/sensors may fail, messages can be lost and the network can be disrupted. Hence, the detection of such failures and malfunctioning is crucial for warnings of critical states, forecasting of disaster scenarios, or preventive measures aiming at avoiding potential disasters, managing crisis or disaster situations.

The main goal of ADMITS is to develop or adapt algorithms, protocols and architectures to enable a decentralized distributed computing environment to provide support for monitoring, failure detection, and analytics in IoT disaster scenarios, considering the characteristics of IoT such as limitation of bandwidth, battery consumption of mobile devices and smart sensors, which impose constraints to communication and dynamics of communication due to mobility of the devices. Specifically, ADMITS research challenges include: (i) To design and evaluate data analytic methods that can be executed over constrained environments in real-time; (ii) To design and evaluate distributed adaptive failure detectors for IoT environments considering communication and battery constraints, the dynamics of the network as well as the relevance (relative importance) of the devices/sensors and the margin of failures that disaster scenarios in question can tolerate; (iii) To design and evaluate an autonomic mechanism for provisioning heterogeneous processing resources for IoT data processing in disaster scenarios; (iv) To design and evaluate a distributed architecture to integrate the previous features including: (1) failure detection, (2) data analytics, and (3) and real-time data stream processing.

The remainder of this paper is structured as follows: Section 2 introduces key concepts of ADMITS. Section 3 describes the proposed methodology. Section 4 briefly discusses related work. Section 5 concludes this paper.

## 2. Outlook of the ADMITS Proposal

ADMITS is based on the expertise of the teams in several complementary areas, including large scale and mobile distributed computing and algorithms, fault tolerance (e.g. adaptive failure detection, replication), dynamic and heterogeneous systems, real-time stream processing and data analytics, large scale data management, self-organized systems, and post-disaster geo-location information gathering. Figure 1 depicts an ADMITS-enabled ecosystem, highlighting underlying infrastructure along with participating technologies.

### Distributed failure detection in IoT environments

IoT's distributed and dynamic nature as well as energy, bandwidth and communication constraints of sensors and devices present new challenges for the conception of an unreliable failure detector. Hence, inspired on the Impact FD, we intend to propose an adaptive failure detector whose implementation is communication-cost efficient, since sending and receiving data is one of the most energy consuming operations on mobile devices and sensors, tackles with mobility of devices and network partition, and dynamically adapts to the changes in relation to failure suspicions and environment.

For instance, approaches such as message combining or inclusion of detection information in application messages should be considered. Moreover, in the current conception of Impact FD, the node relevance (i.e., their *impact*) and the threshold values are statically assigned in the beginning of the monitoring process and do not change. In IoT context, the former should be dynamically re-evaluated based on network instability, node energy consumption or another parameter of the environment that changes over

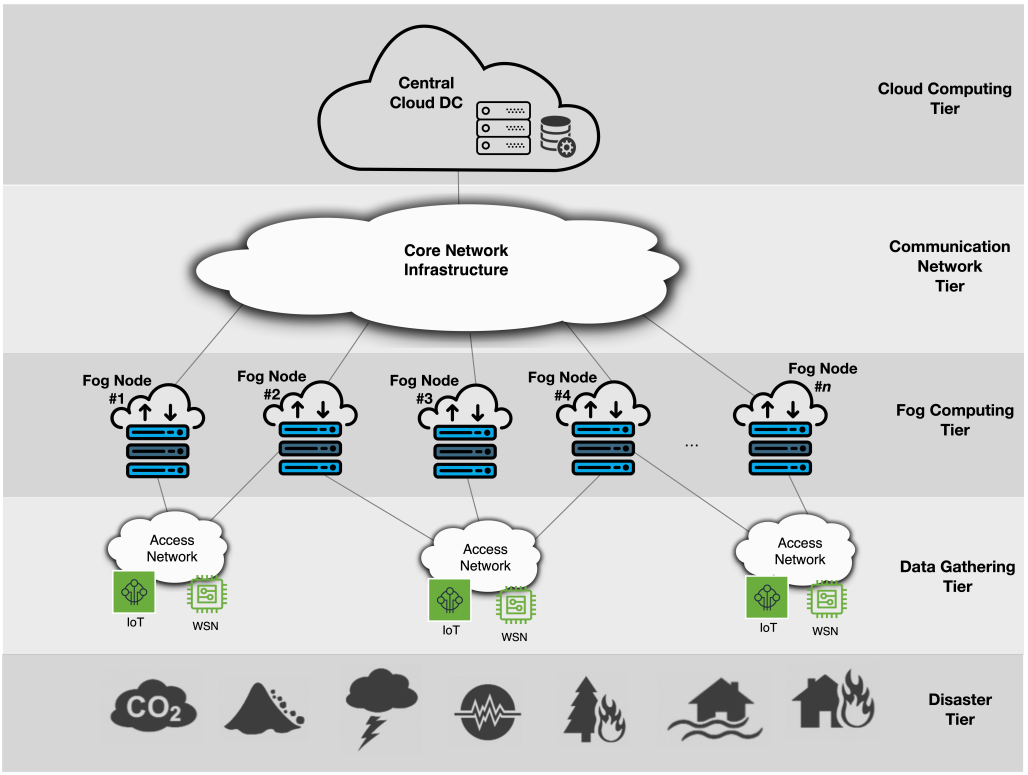


**Figure 1. A prospecting ADMITS-enabled ecosystem.**

time, while the latter should change based on the necessity of stronger or more relaxed monitoring which can vary over time in disaster monitoring management.

Another point is that edge nodes are more stable, robust, and powerful than mobile devices and sensors. Then, instead of a single monitoring node, several edge nodes could have this role and exchange their trust level output providing, therefore, a more scalable and robust failure detection. We can also consider, even if edge nodes are more reliable than IoT, to have a second failure detector (a traditional one [Chandra and Toueg 1996]) in order to monitor edge nodes. In the case of an edge node failure, the tasks performed by the faulty node would be replaced by another edge node.

### Data analytics in IoT and disaster scenarios

On the basis of the dynamic nature of environments subject to disasters, any architectural design must consider that any type of failures is feasible. When it comes to data analytics, key aspects are required in order to make the system robust to failures, such as tolerating cuts in the data pipeline in which sensed attributes are flowing, and seasonality that has potential to modify the set of possible disasters. ADMITS considers investigations from the two above challenges, seeking efficient and robust models that can be placed closer to the disaster areas. By developing a hierarchical architecture, in which data analytics occur at two levels, distributed Fog and centralized Cloud, ADMITS can take advantage of the processing capacities located in each place.

When collecting data from the sensors, ADMITS applies feature selection methods that efficiently run at Fog (reduced processing capacity), not only reducing the amount of data pushed towards the Cloud (network capacity constraints), but mainly provid-

ing fault resilience to such data. For example, under non-disaster conditions we might have several sensors in a given area, producing hundreds of metrics every adjustable time-interval that are fed to our analytic models, but suddenly, a considerable set of sensors are lost due to a disaster, breaking the flow of data produced by them. In order to detect such a scenario, the data analytics software invokes the failure detector (FD) described above. Based on a threshold value and the trust level rendered by the FD, it can take decisions about the confidence degree of IoT and collected data. ADMITS investigates alternatives to amend such effects, maintaining the representativeness of collected data under disaster conditions in order to keep the computed models usable [Shcherbakov et al. 2017, Bacciu 2016].

Another important aspect is related to the maintenance of analytic models, so the accuracy of the models are kept at high levels. In this context, ADMITS investigates incremental methods for handling seasonality of the disaster scenarios, removing some learned concepts from the models in order to improve its accuracy. For example, during the year, we might remove the rainy season feature from the model to include another feature such as dry season, as fires are more susceptible to it.

**Real-time stream processing system**

IoT has experienced a large expansion in the last years due to the miniaturization and cost reduction on the sensors and smart devices such as smartphones enabling continuous sensing on the most diverse environments. Middleware for data analytics is considered a key element in a IoT architecture and stream processing systems is one approach to implement this middleware layer [Buddhika and Pallickara 2016].

Stream processing systems with time-critical requirements and which are also subject to infrastructure failures are two key challenges that the disaster scenario imposes to stream processing systems. In this project, we propose to move processing towards the edge of the network so as to avoid data movement towards the Internet backbone and improve timely response. However, in this context, the stream processing system has to run over constrained devices, prohibiting the use of costly models for supporting adaptation to failures. We expect to use a hierarchical model to approach these issues.

In previous work [Morales et al. 2014], we have tolerated failures using checkpoint techniques, taking into account device mobility and signal strength. In this project, we propose to integrate the failure detection models specially designed for this context to the monitoring phase of the adaptive stream processing system in order to plan the execution of tasks over the available resources in the edge.

**Distributed integrated architecture**

Dealing with failures is a must for any disaster management system. Another requirement is to be able to integrate wireless sensor networks (WSN), IoT, Fog and Cloud computing infrastructures even in presence of failures and communication disruption. Data stream collected and processed from WSN and IoT devices are then processed by data analytic models using available resources in the Fog, aiming at decreasing the volume of data sent to the Cloud. Thus, it is necessary to develop a distributed platform capable of supporting an integrated view of the whole life cycle of a disaster management system and supporting better decision support for people in charge of acting on disaster situations.

Such architecture will integrate the above three system components (data analytics, stream processing, and failure monitor nodes) that will run in Fog nodes. The data analytics software will also communicate with the Cloud.

## 3. Research Methodology

The ADMITS project has as general goal the design of efficient distributed information systems, which support technologies atop IoT and WSN device infrastructures to afford detecting, forecasting, and managing disasters. In order to achieve the main goal of the ADMITS project, we defined a set of specific goals, which are elicited in the following:

1. Failure detection in IoT, where disaster can take place, is crucial since it allows early warning of environment changes, notification of critical situation and forecasting of possible disaster. The failure detector should be adaptive and tailored to the dynamic nature of IoT, its constraints in terms of hardware (e.g. low battery, memory restriction, etc.), communication dynamics, heterogeneity of nodes, and failure tolerance flexibility, also exploiting edge nodes which are more stable.
2. Data analytics algorithms using real-time processing systems should be designed to execute efficiently over restrained IoT devices to provide fast answer in critical pre and post disaster scenario tasks. We propose to explore efficient and robust analytic models that can be placed closer to the disaster areas tolerating breakages on data monitoring due to disasters, and adapting along the year to different seasons while keeping high accuracy operating levels.
3. We propose to explore real-time stream processing systems which are capable of adapting themselves as a middleware to support data analytic in disaster scenarios or environments prone to it. The deployment of processing tasks at the edge of the network may improve response times in IoT-based applications. We propose to integrate a failure detection model in the monitoring phase of the adaptive processing system in order to plan the execution of the tasks over the available resources in the edge.
4. In light of the dynamic nature of the environment, energy power of the devices, the amount of data produced by the IoT, and large scale issues, the development of a distributed architecture to support the integration of the multiple underlying enabling technologies to afford disasters forecasting or managing is a great challenge. Furthermore, the deployment of data analytics models should be optimized according to the capabilities and limitations of the Fog (data streams) and communication with the Cloud (intensive batch processing) computing infrastructures.

We will validate our approach by conducting experiments on the FIT IoT-LAB [Adjih et al. 2015], a large-scale experimental testbed allowing design, development, deployment and testing of innovative IoT applications. We will also adopt the Brazilian DOJOT [Platform 2018] platform for the development of testbeds which allow the monitoring of IoT devices and connecting them to Fog and Cloud platforms for analytics. IoT connectivity within the partners' countries can be achieved using SigFox [SigFox 2018], a networking service provider specialized in IoT connectivity which can support the deployment of our testbeds, integrating the solutions with the DOJOT platform, for example.

A different approach for modeling large-scale IoT systems and exploring fault-tolerance strategies is relying on an Agent based Model (AbM) simulator. Specifically,

each agent can be treated as an individual and independent entity in the environment, accounting for each individual IoT device, Fog machine or Cloud server. Communication is also accounted for by means of local connection (wired/LAN) or long-distance signals. Also, since each entity in the simulation is independently programmable, behaviour of each of the agents can be made to simulate device actions. The UDP team will test this approach using `gro` [Jang et al. 2012, Gutiérrez et al. 2017], a bacterial colony simulator able to handle large amounts of bacteria simultaneously.

## 4. Related Work

Due to the large scope of the project, we group related work in four main categories.

### Distributed failure detection in IoT environments

In distributed systems, detection of crashed devices and network disruptions does not occur in real-time and is not always reliable since, sometimes, it is not possible to know whether a device has really failed, a message has been lost or if the device and/or the network communication are just slow. Proposed by Chandra and Toueg in [Chandra and Toueg 1996], unreliable failure detection (FD) can be seen as an oracle that gives (not always correct) information about node failures (either trusted or suspected). It usually provides a list of nodes suspected of having crashed. It can make mistakes by erroneously suspecting a not crashed process (false suspicion), or by not suspecting a process that has actually crashed. If the FD detects its mistake later, it corrects it.

However, disaster management systems are usually interested in information about the reliability of the IoT monitoring network as a whole and not each individual device (sensor), and can often tolerate a certain degree of failures due, for instance, to redundancy of sensors that perform the same task. Furthermore, devices may be heterogeneous having different importance (relevance) or roles and, thus, their failures may have distinct impact on the system. In [Rossetto et al. 2018], we propose an unreliable failure detector, the Impact FD, where a node monitors a set of nodes (sensors, devices, processes), and the FD oracle of the monitor node outputs its trust with regard to the set of the monitored nodes as a whole and not for each of these nodes. The set is considered "trusted" if it behaves correctly for a specific purpose even in the face of failures, i.e., the current set of monitored nodes that the FD consider not failed are able to maintain the normal monitoring functionality. Furthermore, the Impact FD allows to assign different relevance values (relative importance) to nodes and define a lower bound (threshold) over which the confidence degree on the set of monitored nodes is ensured, offering, therefore, a degree of flexibility for failure and false suspicions.

### Data analytics in IoT and disaster scenarios

The disaster scenarios represent a typical non-stationary data stream [Gama 2010] environment in which the concepts are not static, but they evolve over time. In this scenario, an important phenomenon called concept drift [Gama et al. 2014, Webb et al. 2016] may occur. It can be described as a significant change in the data distribution. This phenomenon is especially challenging when noise is present within the data. In addition, models trained using data from non-disaster scenarios must be able to detect novel occurrences of disasters and update the model in order to incorporate this knowledge. In machine learning, this task is referred to as novelty detection [Faria et al. 2016]. Another related technique

is progressive learning, in which data classification is carried out, and further addition of data is then either associated to existing classes or assigned to an entirely new class [Venkatesan and Er 2016].

### Real-time stream processing system

Previous work [Hidalgo et al. 2017] has shown that autonomic stream processing systems are able to deal with burst of traffic that can be generated in the context of disaster scenarios, adjusting the internals of the systems to the current traffic. This process may enable our system to deal with the complexity of managing data analytics algorithms over environments subject to failures and integrate monitoring, planning, and execution capabilities so as to satisfy some utility goals (e.g., maximize performance, optimize resource usage, guarantees on processing reliability, etc.). In order to achieve good QoS for time-critical applications, such as data analytics for disaster scenarios, several works have been proposed without putting the focus on failure [Cardellini et al. 2018][Cardellini et al. 2017].

### Distributed integrated architecture

The two works closest to our proposal are [Uddin et al. 2016] and [Furquim et al. 2018]. In [Uddin et al. 2016], Uddin et al. propose SCALE2 which engages a multi-network approach to drive data flow from IoT devices to cloud platforms where the analytics are executed. The system is organized in a hierarchical approach for managing a community of IoT devices while focusing on resilience methods that can be employed at different tiers in the hierarchical architecture. The article [Furquim et al. 2018] presents a fault-tolerant three-tier (IoT, Fog, and Cloud) system with data analytics, for the detection and forecasting of flood disasters and the issuing of alerts. Contrary to our approach, that provides a flexible failure detection that tolerates a margin of failures, in those systems fault-tolerance is embedded in the system by anticipating the risk of communication breakdowns and/or the destruction of the nodes. Dynamics of the environment are not taken into account, and data are collected by the fog nodes and analyzed by the Cloud nodes, while in our approach data analytics are performed at fog nodes for communication effectiveness sake.

## 5. Conclusions and Future Work

In this paper, we present the ADMITS proposal and discuss several issues related to the current IoT-based disaster management solutions. With that in mind, the goal of the project is to design and evaluate a distributed architecture to integrate failure detection, data analytics, and real-time data stream processing in IoT disaster scenarios. In future work, we will present results from the architecture implementation which includes experimental validation to demonstrate the feasibility and potential benefits of ADMITS.

## References

Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noël, T., Pissard-Gibollet, R., Saint-Marcel, F., Schreiner, G., Vandaele, J., and Watteyne, T. (2015). FIT iot-lab: A large scale open experimental iot testbed. In *2nd IEEE World Forum on Internet of Things, (WF-IoT)*, pages 459–464.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376.

Bacciu, D. (2016). Unsupervised feature selection for sensor time-series in pervasive computing applications. *Neural Comput. Appl.*, 27(5):1077–1091.

Buddhika, T. and Pallickara, S. (2016). Neptune: Real time stream processing for internet of things and sensing environments. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 1143–1152.

Cardellini, V., Grassi, V., Lo Presti, F., and Nardelli, M. (2017). Optimal operator replication and placement for distributed stream processing systems. *SIGMETRICS Perform. Eval. Rev.*, 44(4):11–22.

Cardellini, V., Lo Presti, F., Nardelli, M., and Russo Russo, G. (2018). Towards hierarchical autonomous control for elastic data stream processing in the fog. In Heras, D. B., Bougé, L., Mencagli, G., Jeannot, E., Sakellariou, R., Badia, R. M., Barbosa, J. G., Ricci, L., Scott, S. L., Lankes, S., and Weidendorfer, J., editors, *Euro-Par 2017: Parallel Processing Workshops*, pages 106–117, Cham. Springer International Publishing.

Chandra, T. D. and Toueg, S. (1996). Unreliable failure detectors for reliable distributed systems. *Journal of the ACM (JACM)*, 43(2):225–267.

Chiang, M. and Zhang, T. (2016). Fog and iot: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6):854–864.

Cisco (2015). Fog computing and the internet of things: Extend the cloud to where the things are. Cisco White Paper.

Faria, E. R., Gonçalves, I. J., Carvalho, A. C., and Gama, J. a. (2016). Novelty detection in data streams. *Artif. Intell. Rev.*, 45(2):235–269.

Furquim, G., Jalali, R., Pessin, G., Pazzi, R. W., Ueyama, J., et al. (2018). How to improve fault tolerance in disaster predictions: a case study about flash floods using iot, ml and real data. *Sensors*, 18(3):907.

Gama, J. (2010). *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, 1st edition.

Gama, J. a., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):44:1–44:37.

Guha-Sapir, D., Hoyois, P., and Below, R. (2016). Annual disaster statistical review 2016: The numbers and trends. Brussels: CRED.

Gutiérrez, M., Gregorio-Godoy, P., Pérez del Pulgar, G., Muñoz, L. E., Sáez, S., and Rodríguez-Patón, A. (2017). A new improved and extended version of the multicell bacterial simulator `gro`. *ACS Synthetic Biology*, 6(8):1496–1508.

Hidalgo, N., Wladdimiro, D., and Rosas, E. (2017). Self-adaptive processing graph with operator fission for elastic stream processing. *Journal of Systems and Software*, 127:205 – 216.

Jang, S. S., Oishi, K. T., Egbert, R. G., and Klavins, E. (2012). Specification and simulation of synthetic multicelled behaviors. *ACS Synthetic Biology*, 1(8):365–374.

Kreps, J. (2014). Questioning the lambda architecture. https://www.oreilly.com/ideas/questioning-the-lambda-architecture.

Morales, J., Rosas, E., and Hidalgo, N. (2014). Symbiosis: Sharing mobile resources for stream processing. In *2014 IEEE Symposium on Computers and Communications (ISCC)*, volume Workshops, pages 1–6.

Platform, C. D. (2018). http://www.dojot.com.br/.

Rauniyar, A., Engelstad, P., Feng, B., and Thanh, D. V. (2016). Crowdsourcing-based disaster management using fog computing in internet of things paradigm. In *2nd IEEE International Conference on Collaboration and Internet Computing, CIC 2016, Pittsburgh, PA, USA, November 1-3, 2016*, pages 490–494.

Ray, P. P., Mukherjee, M., and Shu, L. (2017). Internet of things for disaster management: State-of-the-art and prospects. *IEEE Access*, 5:18818–18835.

Rosas, E., Hidalgo, N., Gil-Costa, V., Bonacic, C., Marin, M., Senger, H., Arantes, L., Marcondes, C., and Marin, O. (2016). Survey on simulation for mobile ad-hoc communication for disaster scenarios. *Journal of Computer Science and Technology*, 31(2):326–349.

Rossetto, A., Geyer, C., Arantes, L., and Sens, P. (2018). Impact fd: An unreliable failure detector based on process relevance and confidence in the system. *Computer Journal, To be published*.

Shcherbakov, M. V., Brebels, A., Shcherbakova, N., Kamaev, V., Gerget, O., and Devyatykh, D. (2017). Outlier detection and classification in sensor data streams for proactive decision support systems. *Journal of Physics: Conference Series*, 803(1):012143.

SigFox (2018). https://www.sigfox.com/.

Siozios, K., Anagnostos, D., Soudris, D., and Kosmatopoulos, E. (2018). *IoT for Smart Grids: Design Challenges and Paradigms*. Power Systems. Springer International Publishing.

Stolpe, M. (2016). The internet of things: Opportunities and challenges for distributed data analysis. *SIGKDD Explor. Newsl.*, 18(1):15–34.

Uddin, M. Y. S., Nelson, A., Benson, K., Wang, G., Zhu, Q., Han, Q., Alhassoun, N., Chakravarthi, P., Stamatakis, J., Hoffman, D., et al. (2016). The scale2 multi-network architecture for iot-based resilient communities. In *Smart Computing (SMARTCOMP), 2016 IEEE International Conference on*, pages 1–8. IEEE.

Venkatesan, R. and Er, M. J. (2016). A novel progressive learning technique for multi-class classification. *Neurocomputing*, 207:310–321.

Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., and Petitjean, F. (2016). Characterizing concept drift. *Data Min. Knowl. Discov.*, 30(4):964–994.

Yang, L., Yang, S., and Plotnick, L. (2013). How the internet of things technology enhances emergency response operations. *Technological Forecasting and Social Change*, 80(9):1854 – 1867. Planning and Foresight Methodologies in Emergency Preparedness and Management.