Pervasive Data Science on the Edge

Lagerspetz, Eemil

2019-11-28

Lagerspetz , E , Hamberg , J , Li , X , Flores , H , Nurmi , P , Davies , N & Helal , S 2019 , ' Pervasive Data Science on the Edge ' , IEEE Pervasive Computing , vol. 18 , no. 3 , pp. 40-48 . https://doi.org/10.1109/MPRV.2019.2925600

http://hdl.handle.net/10138/309039
https://doi.org/10.1109/MPRV.2019.2925600

acceptedVersion

THEME ARTICLE

Pervasive Data Science on the Edge

Eemil Lagerspetz
University of Helsinki

Jonatan Hamberg
University of Helsinki

Xin Li
University of Helsinki

Huber Flores
University of Helsinki

Petteri Nurmi
University of Helsinki

Nigel Davies
Lancaster University

Sumi Helal
Lancaster University

Proliferation of sensors into everyday environments is resulting in a connected world that generates large volumes of complex data. This data is opening new scientific and commercial investigations in fields such as pollution monitoring and patient health monitoring. Parallel to this development, deep learning has matured into a powerful analytics technique to support these investigations. However, computing and resource requirements of deep learning remain a challenge, often forcing analysis to be carried at remote third-party data centres. In this article, we describe an alternative *computing as a service* model where available smart devices opportunistically form micro-data centres that can support deep learning-based investigations of data streams generated by sensors. Our model enables smart homes, smart buildings, smart offices, and other types of smart spaces to become providers of powerful computation as a service, enabling edge analytics and other applications that require pervasive (in-space) decisioning.

INTRODUCTION

Pervasive data science is an emerging research field in the intersection of pervasive computing and data science [1] that builds on the increased proliferation of sensors into everyday environments. These sensors produce large volumes of complex, real-time data streams, which are opening scientific investigations at an unprecedented scale. Examples of the adoption of pervasive computing and sensors abound, with pollution monitoring [2] and patient health monitoring [3] being examples of domains benefiting from pervasive sensor data. Besides scientific benefits, data generated by pervasive computing deployments has large commercial potential. Indeed, an increasing fraction of the big data market – estimated to be over $65 billion worldwide in 2018 [4] – operates on data resulting from pervasive computing deployments.

Parallel to the emergence of pervasive data science, computing has witnessed a paradigm shift towards *big data*. This shift is characterized by computing becoming reliant on ever-increasing volumes of data with datasets containing dozens of terabytes or even exabytes of measurements being increasingly commonplace. Despite significant advances in analysis techniques, including emergence of efficient and easy-to-use deep learning frameworks (e.g., TensorFlow), the massive scale of these datasets necessitates the use of powerful computing infrastructure beyond the capabilities of individual devices. Indeed, most data scientific investigations operate on remote data centres that offer access to powerful large-scale (big data) computing infrastructure. While this suffices for traditional data science, it is insufficient for pervasive data science due to two main limitations. Firstly, data collected through pervasive computing deployments tends to have short time span, making it necessary to analyse the measurements near real-time. With current pricing structures this would be prohibitively costly, resulting in monthly costs that exceed thousands (academic) or even tens of thousands of dollars (commercial). Secondly, sensor data streams are characterized by high data velocity and need for (near) interactive response rate. Satisfying these interactivity requirements requires processing infrastructure to be located close to the data source, rendering remote data centres unsuitable. Even when data centres are located physically close to the end user, they remain distant in networking terms. Indeed, locating data centres close to end users can reduce latency but not eliminate it. Latencies below 100ms [5] are rare, meaning only 10 interactions per second can be achieved. Such latency is insufficient for applications with high interactivity requirements (e.g., augmented reality or other applications operating on interactive video) or large volume combined with high velocity (e.g., autonomous cars, surveillance cameras, or augmented cognition solutions). For example, autonomous cars are estimated to generate data at nearly 1GB/s for making real-time driving decisions. [6] To tailor analysis to the needs of these applications and investigations, computing infrastructure needs to be located close to the devices producing data.

In this article, we propose a *computing as a service* model where *unused computing resources of available smart devices are opportunistically harnessed to form micro-scale data centres* for processing pervasive data science investigations. By locating processing closer to the sensors and by taking advantage of low-latency connectivity technology, such as Bluetooth low energy (BLE), the generated data can be analysed promptly by avoiding the multiple network hops needed to transmit data to cloud data centers. In other words, processing becomes available at the *edge of the network*. While some edge computing models, such as *cloudlets* [7], specifically have targeted processing close to the end user, they are insufficient for pervasive data science as they require dedicated processing hardware in proximity of the sensors. Besides limited incentives for deploying such hardware, individual devices unlikely have sufficient processing power, necessitating dense deployments that can interact with each other – in line with the vision of fog computing [8]. Due to costs associated with this kind of scenario, this is unlikely to happen – at least beyond specialised commercial scenarios (such as wind farms and power grids) where commercial and other benefits (such as security) would drive deployments. Instead, we consider a model that harnesses *idle computing power of available smart devices* for the purpose of pervasive data scientific investigations. The objective of this article is to demonstrate the viability of this model as a potential research direction that can provide the necessary system support for pervasive data science. Indeed, our proposed computing model supports data streams with both high velocity and volume, and provides a mechanism for distributing computing in a way that enables devices to optimize resource consumption. Besides demonstrating the potential of our model, we identify and highlight key research challenges, aiming to act as catalyst for research in this area.

EXAMPLES OF APPLICATION DOMAINS

To illustrate the potential of our approach as a foundation for a wide range of everyday computing scenarios, we highlight potential application domains that can benefit from opportunistically formed smart device micro-data centres. To highlight the range of opportunity, we consider scales from a handful of devices (3-5) to ones involving tens or even hundreds of devices simultaneously.

Smart Homes: The most intuitive example of smart device data centres is a smart home that provides a small number of smart devices (3-5) ranging from personal devices to household appliances, such as smart TVs, smart fridges or even smart washing machines. The combined processing power of these devices suffices for boosting personal computational needs and exceeds those of a dedicated edge device, such as cloudlet, without the need to purchase and deploy a separate device for this purpose. As an example, smart device data centres can run deep learning models for multimodal activity recognition (e.g., audio and motion data). Alternatively, smart home data centres can operate within a larger cluster, e.g., a city-wide virtual data centre consisting of buildings data centres, which in turn consists of individual homes' micro-data centres. The availability of these devices varies. Household appliances typically have idle resources throughout the day whereas other devices, such as smart TVs, may need to be turned on to allow resources to be utilized. Note that even if the devices are used for their primary use, some resources typically remain free, suggesting that devices could perform computations on the background even when operated normally. This is particularly the case for devices with constant power supply.

Smart Offices: Business spaces often are proliferated with smart devices with varying computing capabilities. For example, employees are likely to have, smartphones and laptops, and smart displays, routers and other programmable devices. These devices could be used to support business intelligence operations of the company. This is particularly relevant with start-ups and early-stage SMEs that have limited resources, but where most employees will have one or several smart devices. Note that while this requires operating computations on a heterogeneous pool of devices, it reduces the need to buy expensive in-house infrastructure, pay for leased cloud services, or share data with third party provides. Smart offices represent a larger source of processing power. Besides business intelligence, they would be suitable for tasks such as object recognition and labelling from live surveillance footage, or augmented reality 3D object design.

Smart Campuses: One of the early motivators for our research was to support developing countries where access to high-end computing services is infeasible due to high cost. However, smartphone ownership tends to be high, and hence it would be possible to use devices or employees and students for supporting campus-wide computing jobs. Similarly, in developed countries, many campuses offer laptops or other devices to their students. Requiring a fraction of computing time of the laptop as a condition to receive the device could thus be used to augment the computational resources that are available on campus. Examples of suitable campus-wide computing jobs include weather models, air quality sensing, and diverse neural network use cases.

Sporting Events: Large events with thousands of people, such as sporting events, concerts, demonstrations, and political campaigns have the potential of a thousand devices in the same location. This could be used for e.g. super-resolution imaging [9], augmenting the resolution of videos taken at the event for the benefit of all participants. Investigations that support safety and well-being of the attendees, such as air quality modelling, could also take advantage of this enormous but short-lived concentration of computing power.

RESEARCH CHALLENGES

Novel system architectures that can support high volume and high velocity analysis, while supporting offloading or other techniques for reducing resource drain of devices producing data, has been identified as one of the key system design research challenges for pervasive data science [1]. Opportunistically formed smart device data centres address this challenge without requiring any additional infrastructure. Enabling this vision, however, comes with its own research challenges. We next highlight and contextualize these challenges relative to current solutions.

Data Volume and Velocity: Large volumes of real-time data with high sampling rates that require immediate analysis are produced by smart and other sensor-enabled devices. Examples range from real-time object recognition to sensor analysis for autonomous cars and cognitive assistance. The capabilities of a smart device, or even a dedicated edge/fog node, unlikely suffice for these tasks. For example, object recognition in 30-60 frame per second (FPS) 4K video streams is difficult for an individual device. Handling these streams requires mechanisms that can distribute tasks between participating devices while taking velocity and volume into consideration. For example, high volume requires contributing devices to be sufficiently powerful to ensure communication bandwidth does not become an issue. When volume is smaller, but velocity is high (e.g., audio or motion data), distribution becomes more efficient and devices with less computational power can be considered. Current frameworks for data science, are insufficient as they do not consider different characteristics of the tasks – or requirements of individual devices, such as ensuring they do not run out of battery while processing computing tasks.

Distribution of Machine Learning: Training deep learning models requires significantly more resources than running pre-trained models. While model training could be done directly on the device, e.g., using distributed deep learning tools, [10] the main benefits of our approach result from distributing *inference* as this allows local processing of high velocity data streams. Distribution of inference can be achieved by partitioning input measurements and running the model for different partition on each device (followed in our experiments) or separating data streams and giving each device one sub-stream.

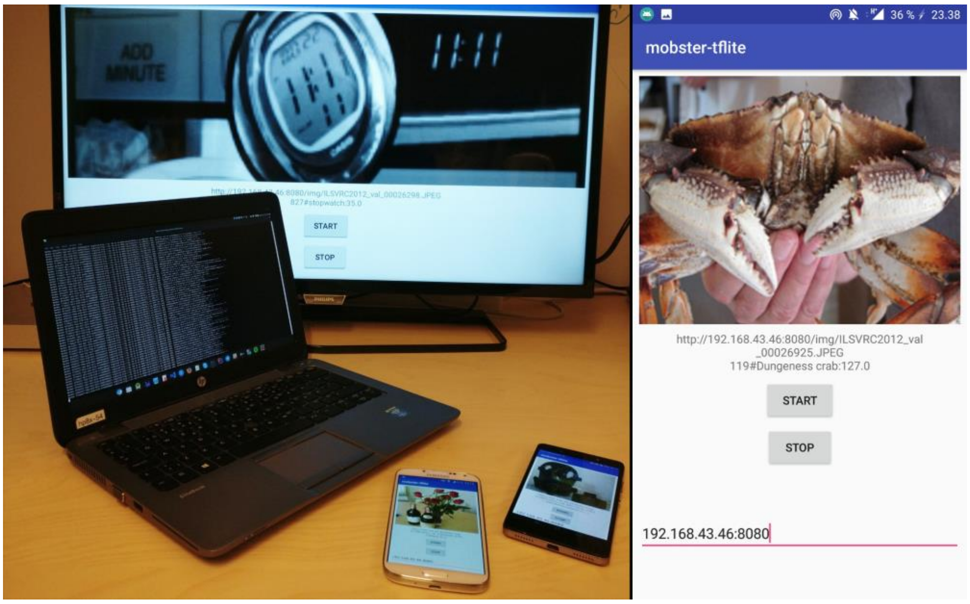
Device Heterogeneity: Computing infrastructures commonly employed in data science rely on high-power devices with large memory capacity. This contrasts with smart devices which tend to have powerful computational resources but are limited in battery, memory, and I/O resources. While some research has investigated using smart devices (especially smartphones) as part of larger distributed computing infrastructure (see Related Work sidebar), these solutions simply see smart devices as constrained nodes without optimizing task allocation according to characteristics of the devices. Addressing device heterogeneity requires solutions that can embrace this diversity of smart devices and best take advantage of their individual capabilities instead of considering it as limitation. For example, computation can be distributed to most powerful devices, while other devices can provide networking or storage capabilities. However, supporting this requires programming models that offer finer-grained control of task distribution amongst participating devices. Additionally, smart device programming models that support multi-threading and continuous background processing need to be available for a wide range of smart devices. For example, smart fridges and smart TVs would need to allow applications that can operate even when the devices are in standby without requiring the user to launch a dedicated app.

Incentives: Would people contribute computing resources is a common critique aimed at solutions that bring computing onto smart devices. While valid concern, it is not unique to computing. Similar concerns have been raised, e.g., in crowdsourcing and opportunistic networking, and solutions from these fields could be adopted. One solution has been to rely on micro-payments to users contributing resources. Indeed, research has shown that users would be willing to contribute parts of their computational power in exchange for money [11]. Another alternative would be enforcement. For example, a company could require employees to allow a fraction of their device's computational power for processing tasks during work hours – or a smart city could rent smart homes with reduced price in exchange for access to computational power offered by smart devices installed into the home. We note that simply motivating users to offer resources is insufficient as this is likely to affect other stakeholders. For example, coffee shops, airports and other venues might suffer from higher electricity costs when users charge their devices at the same time as they offer computational support for payments. Thus, solutions that look beyond concerns of individual users need investigation.

Security: Since performing shared computation implies disclosing valuable information between devices, it is critical for devices to implement mechanisms that ensure that a shared computation does not disclose the result to others, but just to the one that initiated the data computation request. Similarly, for owners of the devices that contribute resources, it is essential that access to computational resources does not allow malicious access to data on their device. Traditional security solutions, such as sandboxing, provide partial solution for securing the client devices whereas file systems offering time-limited access to data helps to reduce risks of malicious use for the task initiators.

CASE STUDY: OBJECT REGONITION

We demonstrate the feasibility and potential of smart device data centres through a case study of deep learning-based object recognition from real-time video. Examples of real-world applications that benefit from real-time capability to detect objects range from visual surveillance to activity recognition, and crowd monitoring.



http://192.168.43.46:8080/img/ILSVRC2012_val_00026925.JPEG
119#Dungeness crab:127.0

192.168.43.46:8080

FIGURE 1: Left: Object recognition experiment running on Smart TV and two smartphones, coordinated by the laptop. Right: Screenshot of a smartphone running the experiment.

Experiment and Prototype Description

We run deep learning experiments using the MobileNet [12] model to recognize objects from a video feed. We implement a proof-of-concept prototype that uses a master/worker topology, analogously to current big data frameworks, where each node in the topology is a device. A master device is elected to coordinate the execution of **tasks**, parts of the overall job. The remaining devices act as workers that contribute to the overall job by running one or more tasks. In our prototype, each device has a background service with a persistent low-power connection. Smart devices with excess performance register resources they wish to make available by communicating them to the master. Devices can allocate a fraction of their maximum computing power. Smaller values can be used to reduce battery life effects for devices that are powered by a battery, and to limit performance impact for devices currently in use. To handle devices leaving or becoming unresponsive, speculative execution can be used, submitting each task to multiple devices, and the result is only accepted from the first device to complete the task; duplicates of the task are cancelled as the first result is received. This incurs an overhead of extra task scheduling and cancellation on multiple devices, so speculative execution is usually reserved for situations where a task has already been running for longer than expected on a computing device.

Implementation

We developed an Android application that performs the object recognition on images distributed by a master node responsible for coordinating the task of annotating the entire video feed. For deep learning, we use TensorFlow Lite (https://www.tensorflow.org/lite/). The application, including cluster formation and task distribution functionality, consisted of 1,000 lines of code (Android and NodeJS). We use a subset of 1,000 images with a resolution of 224x224 from the ImageNet Large Scale Visual Recognition Challenge (ILSVRC2012) [13] to benchmark the performance of the system. Note that our goal is not to improve accuracy of object recognition, but to demonstrate the benefits of using opportunistic smart device data centres for a real-world task.

We utilize a pre-trained and quantized Convolutional Neural Network (CNN) model, MobileNet_1_0.5.25_224. Each device concurrently calculates and returns results for the object detection tasks they receive from the master. We conduct the experiment on a cluster of 1-4 smart devices and a laptop, which serves as a master node that distributes image paths among the smart devices contributing to the experiment. We use a simple round-robin algorithm for distributing the images. The experiment setup is shown in Figure 1. As smart devices, we consider a Philips Smart TV and three smartphone models: Samsung Galaxy S4, OnePlus 3, and Huawei Honor 7. To provide context for our results, we repeat the experiment on medium to high range Amazon EC2 virtual machines. The choice of master device is arbitrary and could as well be any other network enabled device, such as a smartphone. A limited queue of tasks minimizes idle time as well as waiting for tasks to complete on slower devices. The master continues to distribute new tasks to computing nodes until the job is complete.

Results

Results of our experiments are summarized in Table 1. The columns correspond to the name of the device, or devices, the total time taken for processing 10,000 images, and the speed in frames per second for image recognition. We use the FPS of the cluster as our main evaluation metric since it affects both how fast the analysis can operate and how interactive the resulting application is.

Using more than one device always improves processing speed, but with slower devices (S4 and the smart TV) the combined speed in FPS is close to the sum of individual speeds. When more powerful devices were included, there is more variation; adding the TV to three powerful devices did not improve execution time, even though running the experiment with the TV and any other smart device significantly improved execution speed from that of individual devices. In the best case, performance improves from 37 FPS (on Honor 7) to 74 FPS, i.e., a two-fold increase. This performance is well within interactivity requirements and which would support higher resolution images. In other words, we can both improve interactivity and support higher velocity and volume of data than what individual devices can handle on their own.

Compared to Amazon EC2 VMs, smart devices perform comparably or even slightly better than the VMs. Performance of the top two smart devices was better than a medium Amazon EC2 VM ($0.0416 per hour), and the fastest smart device was only 35% slower than a high-end CPU-only dedicated VM (c5.2xlarge: $0.34 per hour). The dedicated CPU-only VM was equalled and surpassed by two-device configurations (54 and 50 FPS vs 51). GPU-enabled VMs were required to get comparable performance to smart device clusters (p3.2xlarge: $3.06 per hour) and surpass it (p3.8xlarge $12.24 per hour). Running the weaker GPU-enabled VM for just seven days would result in a usage bill of over $500, enough to purchase at least one smart device of comparable power. Results for the cloud deployment were calculated by having all image data deployed in the cloud, i.e., latency was ignored and performance of interactive applications on VMs would be even worse.

# Devices	Device Model(s)	Time (ms)	Speed (FPS)
1	Philips Smart TV (TV)	702,999	14
1	Samsung Galaxy S4 (S4)	491,018	20
1	OnePlus 3 (OP3)	416,327	24
1	Huawei Honor 7 (Honor7)	266,939	37

2	Honor7 + TV	291,159	34
2	OP3 + TV	243,926	41
2	S4 + TV	226,668	44
2	Honor7 + S4	226,262	44
2	OP3 + S4	201,348	50
2	OP3 + Honor7	182,736	54
3	Honor7 + S4 + OP3	146,658	68
4	Honor 7 + S4 + OP3 + TV	135,644	74
Amazon EC2: t3.medium VM		397,563	25
EC2: c5.2xlarge (CPU only)		197,365	51
EC2: p3.2xlarge (without GPU)		287,364	35
EC2: p3.8xlarge (without GPU)		265,634	38
EC2: p3.2xlarge (with GPU)		132,548	75
EC2: p3.8xlarge (with GPU)		111,329	90

TABLE 1: Image recognition time for 1,000 images and speed in FPS of different devices and clusters of devices.

Performance in Real World Scenarios

Environment	Devices	FPS	Investigations
Smart home	3-5	42 – 185	0.7 – 3.08
Smart office	10 – 50	140 – 1,850	2.33 – 30.83
Smart campus	100 – 500	1,400 – 18,500	23.33 – 308.33
Sporting event	500 – 2,000	7,000 – 74,000	116.67 – 1,233.33

TABLE 2: Number of simultaneous investigations of high data rate streams supported in various environments. Investigations refers to number of cameras that can be processed and annotated at 60FPS.

We next consider practical feasibility by extrapolating the results of the investigation onto representative examples of device clusters that would be available in real world scenarios. We consider four representative examples of application domains: smart home, smart office, smart campus, and sporting event. The number of smart devices available and the resulting number of high data rate streams (Investigations) supported is estimated in Table 2. Generally, the computational capacity of existing data centres is limited by available on-site infrastructure, e.g., number of servers. On the other hand, small-scale data centres created with our system can dynamically augment their computational power to rival or exceed the level of processing of existing data centre deployments (even for a short span of time). For instance, a cluster of 100 devices can easily be formed to provide computational services during the lunch hour at a university campus, or a cluster of 1,000 devices or more can be formed for more than one hour during sports events or concerts at stadiums. This suggests that the processing of tasks can be accelerated spontaneously when a powerful cluster is formed and the processing of tasks can be scheduled based on complexity, such that tasks are processed depending on the amount of opportunistic computational power that is formed by devices throughout of the day.

SIDEBAR: RELATED WORK

Cloudlets, defined as a resource-rich and trusted piece of infrastructure located in proximity to end users [7], were among the first architectural models serving as inspiration to current edge computing initiatives. In the cloudlet model, dedicated computational infrastructure, e.g., laptops, displays or base stations, in fixed locations, e.g., restaurants or kiosks, are used as cloudlets that support end devices by augmenting their processing capabilities. The main benefit of a cloudlet is to provision services to users for low latency applications, which allows end devices to improve their performance and battery life. A cloudlet is static by nature. Thus, it suffers from scalability (deploying physical hardware) and maintainability (upgrading hardware). Since a dense cloudlet deployment is required to provide transparent support to devices in the wild, cloudlets solutions have not been adopted in existing architectures. However, the vision of cloudlets has been extended as infrastructure at the *edge* of the network that can improve usability and user experience of remote applications. Cloudlets [7] have been investigated as promising solutions that seek to support data analytics systems with critical response time requirements [5], e.g., video surveillance, health monitoring, and autonomous cars. While cloudlets have evolved during almost a decade, the key issues listed above have not been addressed. Since processing capacity of smartphones has increased considerably, several works have studied the feasibility of using them as edge infrastructure. Table 3 provides a qualitative comparison of frameworks that provide computational support using smart devices of users. The table shows the type of candidate infrastructure that can be used for processing support (*Co-located candidate infrastructure*). Most of the work relies on smartphones for opportunistic computation [14,16]. Verbelen et al [15] used ad-hoc devices available in the environment, such as laptops, and servers. Contrary to our approach which allows fine-grained control over resource use, existing approaches tightly integrate resource-constrained devices with computing infrastructure, resulting in devices having limited control over their resource usage (*Disruption of operations*).

In terms of computational infrastructure capacity, we can observe that most works rely on small cluster sizes (*Formation scale size*). Contrary to previous work, we explore the use of micro data centres that can be opportunistically created by merging the resources of smart devices. Our approach provides a flexible way to augment and upgrade processing capabilities on demand based on dynamic availability of devices. Since our approach requires no fixed hardware installation, it overcomes the scalability issues of deploying computing facilities close to the user. Moreover, since the computational capabilities of the micro data centre can be increased dynamically, we are able to process data intensive tasks on the edge using sophisticated models for deep learning (Deep learning support).

Frameworks	Co-located candidate infrastructure	Deep learning support	Formation scale size	Disruption of operations
Dynamic Cloudlets[15]	Ad-hoc devices	No	Small	Yes
Pocket Cloudlets[16]	None	No	Same device	Yes
HyMobi[11]	Smartphones	No	Small	Yes
FemtoClouds[14]	Smartphones	No	Small	Yes
Our work	Smart devices	Yes	Large	No

TABLE 3: Qualitative comparison of opportunistic computing frameworks using nearby infrastructure to support the user's activities.

CONCLUSIONS

Smart devices are increasingly powerful computing devices that can be exploited on the edge of the network for collective processing. This paper highlighted challenges and opportunities that arise when using smart devices to support pervasive data science. We demonstrate the feasibility and potential of this idea through an example use case of deep learning-based object recognition on a data centre consisting of a laptop and up to 4 smart devices. Our platform and experiments highlight how performing pervasive data science on the edge can provide a cost-effective alternative to remote infrastructure, particularly for tasks characterized by high volume and velocity, but low I/O requirements. Indeed, comparison to cloud computing infrastructure demonstrated that clusters of smart devices can provide comparable, or even better, computational performance with a fraction of the cost and better response rate. This suggests that smart device data centres would be a feasible solution for applications with both high volume and velocity, or that have strong interactivity requirements. Besides demonstrating the feasibility of this paradigm, we also identified key research challenges and application scenarios for opportunistic data centres formed by smart devices.

ACKNOWLEDGEMENTS

This research is partially funded by Academy of Finland grants 296139, 297741, 303825, and 317875.

REFERENCES

1. Davies, N. A. J., & Clinch, S. (2017). Pervasive Data Science: New Challenges at the Intersection of Data Science and Pervasive Computing. *IEEE Pervasive Computing*, *16*(3), 50-58.
2. Shaban, K. B., Kadri, A., & Rezk, E. (2016). Urban air pollution monitoring system with forecasting models. *IEEE Sensors Journal*, *16*(8), 2598-2606.
3. Abawajy, J. H., & Hassan, M. M. (2017). Federated internet of things and cloud computing pervasive patient health monitoring system. *IEEE Communications Magazine*, *55*(1), 48-53.
4. SNS Telecom & IT (2018). The Big Data Market: 2018 - 2030 - Opportunities, Challenges, Strategies, Industry Verticals & Forecasts. *NXTGEN Reports*. Online: https://www.nxtgenreports.com/market-research-reports/the-big-data-market-2018-2030-opportunities
5. Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, *50*(1), 30-39.
6. Luckow, A., Kennedy, K., Manhardt, F., Djerekarov, E., Vorster, B., & Apon, A. (2015, October). Automotive big data: Applications, workloads and infrastructures. In 2015 IEEE International Conference on Big Data (Big Data) (pp. 1201-1210). IEEE.
7. Satyanarayanan, M., Bahl, V., Caceres, R., & Davies, N. (2009). The case for VM-based cloudlets in mobile computing. *IEEE pervasive Computing*.
8. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (pp. 13-16). ACM.
9. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., & Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1874-1883).
10. Sergeev, A., & Del Balso, M. (2018). Horovod: fast and easy distributed deep learning in TensorFlow. arXiv:1802.05799.
11. Flores, H., Sharma, R., Ferreira, D., Kostakos, V., Manner, J., Tarkoma, S., Hui, P., & Li, Y. (2017). Social-aware hybrid mobile offloading. *Pervasive and Mobile Computing*, *36*, 25-43.
12. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
13. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211-252
14. Habak, K., Ammar, M., Harras, K. A., & Zegura, E. (2015, June). Femto clouds: Leveraging mobile devices to provide cloud service at the edge. In *2015 IEEE 8th International Conference on Cloud Computing (CLOUD)* (pp. 9-16). IEEE.
15. Verbelen, T., Simoens, P., De Turck, F., & Dhoedt, B. (2012, June). Cloudlets: Bringing the cloud to the mobile user. In *Proceedings of the third ACM workshop on Mobile cloud computing and services* (pp. 29-36). ACM.
16. Koukoumidis, E., Lymberopoulos, D., Strauss, K., Liu, J., & Burger, D. (2011, March). Pocket cloudlets. In *ACM SIGPLAN Notices* (Vol. 46, No. 3, pp. 171-184). ACM.

ABOUT THE AUTHORS

Eemil Lagerspetz is a Docent and an Academy of Finland Postdoctoral Researcher in Department of Computer Science at University of Helsinki. He completed his PhD in Computer Science at the University of Helsinki in 2014. His research interests include large-scale data analysis, mobile computing, and energy efficiency. Contact him at eemil.lagerspetz@cs.helsinki.

Jonatan Hamberg is a Research Assistant working in the Department of Computer Science at University of Helsinki. His research interests include large-scale distributed systems, big data processing pipelines, and ubiquitous sensing. Contact him at jonatan.hamberg@cs.helsinki.fi.

Xin Li is a Research Assistant and MSc student working in the Department of Computer Science at University of Helsinki. His research interests include deep learning and its applications to mobile and wearable sensing. Contact him at xin.li@helsinki.fi

Huber Flores is a Docent and an Academy of Finland Postdoctoral Researcher in Department of Computer Science at University of Helsinki. He completed his PhD in the Institute of Mathematics and Computer Science at the University of Tartu in 2015. His research focuses on mobile and pervasive computing, distributed systems, and mobile cloud computing. Contact him at huber.flores@helsinki.fi

Petteri Nurmi is an Associate Professor at the Department of Computer Science, University of Helsinki. His research is in the area of pervasive computing and data science, and is characterized by a focus on research questions surrounding the collection, analysis and use of sensor data. Nurmi received a PhD in computer science from University of Helsinki. Contact him at petteri.nurmi@helsinki.fi.

Nigel Davies is a Distinguished Professor in the School of Computing and Communications and co-director of the Data Institute at Lancaster University. His research is in the area of pervasive computing and is characterized by an experimental approach involving large-scale deployments of novel systems. Davies received a PhD in computer science from Lancaster University. Contact him at n.a.davies@lancaster.ac.uk.

Sumi Helal is Professor and Chair in digital health with the School of Computing and Communications, and with the Division of Health Research, Lancaster University. His research interests include pervasive systems, the Internet of Things, smart spaces, with applications to digital health and assistive technologies for successful aging and independence. He received the Ph.D. degree in computer sciences from Purdue University, West Lafayette, IN, USA. Contact him at s.helal@lancaster.ac.uk