AUTONOMOUS HEALTH MONITORING AND ASSISTANCE SYSTEMS USING IOT

EDITED BY: George Azzopardi, Dimka Karastoyanova, Marco Aiello and Christos N. Schizas

PUBLISHED IN: Frontiers in Robotics and AI

frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88966-752-9
DOI 10.3389/978-2-88966-752-9

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: frontiersin.org/about/contact

AUTONOMOUS HEALTH MONITORING AND ASSISTANCE SYSTEMS USING IOT

Topic Editors:
George Azzopardi, University of Groningen, Netherlands
Dimka Karastoyanova, University of Groningen, Netherlands
Marco Aiello, University of Stuttgart, Germany
Christos N. Schizas, University of Cyprus, Cyprus

Citation: Azzopardi, G., Karastoyanova, D., Aiello, M., Schizas, C. N., eds. (2021). Autonomous Health Monitoring and Assistance Systems using IoT. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88966-752-9

Table of Contents

frontiers in Robotics and AI

EDITORIAL
published: 18 March 2021
doi: 10.3389/frobt.2021.611352

Editorial: Autonomous Health Monitoring and Assistance Systems With IoT

George Azzopardi[1]*, Dimka Karastoyanova[1], Marco Aiello[2] and Christos N. Schizas[3]

[1]Information Systems Group, Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Groningen, Netherlands, [2]Department of Service Computing, Institute of Architecture of Application Systems, University of Stuttgart, Stuttgart, Germany, [3]Department of Computer Science, University of Cyprus, Nicosia, Cyprus

Keywords: autonomous health monitoring, elderly, internet of things, early detection and prevention, privacy and scalability

Editorial on the Research Topic

Autonomous Health Monitoring and Assistance Systems with IoT

OPEN ACCESS

Edited by:
Kushal Mukherjee, United Technologies Research Center, Ireland

Reviewed by:
Eric Wade, The University of Tennessee, Knoxville, United States

****Correspondence:***
George Azzopardi
g.azzopardi@rug.nl

Specialty section:
This article was submitted to Biophysics, a section of the journal Frontiers in Robotics and AI

Received: *28 September 2020*
Accepted: *15 February 2021*
Published: *18 March 2021*

Citation:
Azzopardi G, Karastoyanova D, Aiello M and Schizas CN (2021) Editorial: Autonomous Health Monitoring and Assistance Systems With IoT. Front. Robot. AI 8:611352. doi: 10.3389/frobt.2021.611352

INTRODUCTION

The sustainability of the current healthcare system is being challenged by the growing percentage of the aging population. In addition to the financial sustainability entailed by such a trend, other challenges include the long delays in servicing patients, the consequent late detection of serious health issues, and the necessity of hospitalization. Despite certain risks, the majority of elderly people prefer to age in their own homes. As a matter of fact, studies show that elderly people who choose to keep living independently have longer life expectancies than those who join elderly homes. All these put together emphasize the need to develop technological solutions that autonomously monitor and enhance the well-being of the elderly in their homes.

The uptake of the Internet of Things (IoT) opens new opportunities for how technology can assist people in improving their health and well-being along with improving the cost-effectiveness and quality of health and social services. These technological advancements result in various low-cost sensory equipment that benefit the healthcare of the aging population. Such technological innovation contributes to the development of applications, such as the administration of medication, voice command technologies, telemedicine, and others based on artificial intelligence. In particular, machine learning and predictive analysis combined with IoT will play an important role in the early detection of suspicious signs that, if left untreated, can lead to mobility, mental, and cognitive issues. Other applications may aggregate the high frequency, messy, and intermittent data. Such data can then be incorporated in the Electronic Health Record. To enhance privacy, such records can potentially be based on Blockchain platforms and shared with healthcare professionals. Concerns such as scalability, security, and systems interoperability need to be dealt with urgently in order to achieve sustainable healthcare systems.

The identification of the latest developments in this highly interdisciplinary field of research is the theme of the present Research Topic, that is, autonomous health monitoring and assistive systems with IoT. It aims at covering two main aspects of the topic: information systems and artificial intelligence. The former covers the advancement with respect to IoT-related technologies and information systems in terms of scalability, security, and interoperability, while the latter covers novel approaches and applications based on machine and deep learning.

PAPERS INCLUDED IN THIS RESEARCH TOPIC

Wang et al. review automatic elderly fall detection systems from the point of view of data collection, data transmission, sensor fusion, data analysis, security, and privacy. They point out that one of the biggest challenges in this field is the collection of a data set with realistic instances of elderlies' falls. Moreover, it turns out that while various approaches have been proposed, most of them rely on single sensors and work offline. They speculate that better performance is likely to be achieved by focusing on fusing the data from various visual and non-visual portable sensors.

Schiza et al. survey the use that has been made of virtual reality techniques for neurological rehabilitation. Dementia, stroke, spinal cord injury, Parkinson's, and multiple sclerosis are the diseases considered in the review. Such diseases are the ones that appear most promising for virtual reality applications and have been the most investigated in the literature. The outlook that emerges from the research work of Schiza and colleagues is positive with clear signs of virtual reality being an effective, low-cost, and scalable support for rehabilitation.

Turečková et al. propose a convolutional neural network augmented by deep supervision and attention gates for the segmentation of abdomen computed tomography (CT) images. The system is highly beneficial for medical experts and helps them toward better diagnosis of various pathologies. By means of extensive experiments they conclude that their proposed methodology achieves a reliable organ and tumor segmentation from CT scans. They report state-of-the-art performance on various segmentation tasks. Notable is the increase in precision of tumor segmentation.

Cappiello et al. propose a data model suited for modern health monitoring systems where patient data is generated in the periphery of the network. This is crucial for monitoring patients who are not hospitalized. In the model, the patient-generated data is assessed by quality metrics that are context dependent. The proposed model is assessed using actual user data regarding daily physical activities of healthy young people.

CONCLUSION

The articles in this special issue show significant progress toward establishing interdisciplinary research. In particular they encourage interdisciplinary efforts coming from the fields of information systems, distributed systems, artificial intelligence, machine learning, deep learning, software architectures, image processing, virtual reality, monitoring of health, and well-being, among others. The findings presented in these articles also highlight the potential for future advances in each individual field of research as well as at the interface of all relevant research disciplines. If future research places the emphasis on how to serve the healthcare sector and, most importantly, the patients through innovative AI-driven applications, we envision enormous leaps and accumulation of scientific results, software systems, and data. Combined together, they will have a significant societal impact and scientific relevance.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Robotics and AI

SYSTEMATIC REVIEW
published: 16 October 2019
doi: 10.3389/frobt.2019.00100

Virtual Reality Applications for Neurological Disease: A Review

Eirini Schiza[1*], *Maria Matsangidou*[1], *Kleanthis Neokleous*[1] *and Constantinos S. Pattichis*[1,2]

[1] *Research Centre on Interactive Media, Smart Systems and Emerging Technologies (RISE), RISE Limited (RISE), Nicosia, Cyprus,* [2] *eHealth Laboratory, Department of Computer Science, University of Cyprus, Nicosia, Cyprus*

Recent advancements in Virtual Reality (VR) immersive technologies provide new tools for the development of novel and promising applications for neurological rehabilitation. The purpose of this paper is to review the emerging VR applications developed for the evaluation and treatment of patients with neurological diseases. We start by discussing the impact of novel VR tasks that encourage and facilitate the patient's empowerment and involvement in the rehabilitation process. Then, a systematic review was carried out on six well-known electronic libraries using the terms: "Virtual Reality AND Neurorehabilitation," or "Head Mounted Display AND Neurorehabilitation." This review focused on fully-immersive VR systems for which 12 relevant studies published in the time span of the last five years (from 2014 to 2019) were identified. Overall, this review paper examined the use of VR in certain neurological conditions such as dementia, stroke, spinal cord injury, Parkinson's, and multiple sclerosis. Most of the studies reveal positive results suggesting that VR is a feasible and effective tool in the treatment of neurological disorders. In addition, the finding of this systematic literature review suggested that low-cost, immersive VR technologies can prove to be effective for clinical rehabilitation in healthcare, and home-based setting with practical implications and uses. The development of VR technologies in recent years has resulted in more accessible and affordable solutions that can still provide promising results. Concluding, VR and interactive devices resulted in the development of holistic, portable, accessible, and usable systems for certain neurological disease interventions. It is expected that emerging VR technologies and tools will further facilitate the development of state of the art applications in the future, exerting a significant impact on the wellbeing of the patient.

Keywords: virtual reality, head-mounted display (HMD), fully-immersive systems, neurorehabilitation, review – systematic

OPEN ACCESS

Edited by:
Marco Aiello,
University of Stuttgart, Germany

Reviewed by:
Le Li,
First Affiliated Hospital of Sun Yat-sen University, China
Dirk Reiners,
University of Arkansas at Little Rock, United States
Ando Emerencia,
University of Groningen, Netherlands

Correspondence:
Eirini Schiza
ischiza@cs.ucy.ac.cy

Specialty section:
This article was submitted to Virtual Environments, a section of the journal Frontiers in Robotics and AI

Received: *28 June 2019*
Accepted: *26 September 2019*
Published: *16 October 2019*

Citation:
Schiza E, Matsangidou M, Neokleous K and Pattichis CS (2019) Virtual Reality Applications for Neurological Disease: A Review. Front. Robot. AI 6:100. doi: 10.3389/frobt.2019.00100

INTRODUCTION

In recent years, Virtual reality (VR) technology has gained recognition as a useful tool for cognitive research, evaluation, and rehabilitation. A relatively new and a less explored area of VR applications is rehabilitation, helping patients who have lost some of their physical, and/or cognitive abilities to regain these. VR systems allow users to interact in various sensory environments and to obtain real-time feedback on their performance without exposing them to risks while using computer technology. The simulated environments offered via VR technology make it possible for patients to participate in activities in settings and environments like those encountered in real life. In addition, VR tools can be used to record accurate measurements of the user performance and to deliver greater therapeutic stimulation to users.

Some VR applications used in healthcare are for easing pain, anxiety, and distraction where the patient can find himself in an environment of their preference. These applications can provide better mental health and finer quality of life to the patient. Other VR applications are used for cognitive training and patient can work their cognitive abilities by playing a game, while also integrating physical excise aspects. Finally, one of the most complex application solutions in healthcare with VR are physical and neurological rehabilitation. These applications provide functional goals programmed into the virtual reality interactive games, and patients will be able to have a much more fun and engaging therapy experience that will help them rebuild their neurological pathways and inevitably give them the exercise and workout they need. Some examples of these applications can be driving assessment after brain injury where the patient tries to regain his ability to drive. This example can help the patient for his cognitive, motor, and sensory factors. Another common application is the virtual classroom scenario which consists a standard rectangular classroom environment containing desks, a teacher, a blackboard, a side wall with large windows etc. Within this scenario, children's attention performance can be assessed while a series of typical classroom distracters are systematically controlled and manipulated within the Virtual Environment (VE) (Weber, 2019).

Although the use of VR applications is increasing, to the best of our knowledge no systematic review has investigated the use of consumer-oriented fully-immersive VR applications in neurorehabilitation in the past few years along with their effect of these on cognition. To address this gap, the present review examines emerging VR applications developed for the evaluation and intervention of patients suffering from certain neurological diseases.

There are three types of VR systems (Ma and Zheng, 2011): (i) Non-immersive VR systems, is a desktop computer based 3D graphical system which allows the user to navigate the VE that is displayed on a computer screen, typically with the keyboard and the mouse; (ii) Semi-immersive systems project the graphical display onto a large screen, and may rely on some forms of gesture recognition system to implement more natural interactions; (iii) Fully-immersive systems in which the users' vision is fully enveloped, creating a sense of full immersion via a head-mounted display (HMD).

Consumer-oriented fully-immersive VR technologies have advanced quite significantly in the past five years (**Table 1**). These new affordable immersive VR technologies could provide an ideal solution for real clinical settings (Anthes et al., 2016 and Matsangidou et al., 2017). Affordable hardware and open source software prescribe the resources needed to introduce new VR applications. These concepts have successfully managed to address past problems and limitations especially regarding the level of immersiveness and user's interaction in VR applications (**Figure 1**).

Wireless HMDs, haptic input devices, virtual sensory vests omnidirectional treadmills, accurate, and precise tracking systems and optical scanners for gesture-based interaction are nowadays considered to be among the most prominent trends in the field of VR (Anthes et al., 2016). Importantly, most of these technologies incorporate precise and robust sensory data acquisition that can be used in a wide range of applications including medicine, sports training, education, and physical/mental rehabilitation.

The objective of this paper was to carry out a systematic review of emerging VR applications developed over the last 5 years, covering selected neurological diseases. More specifically, this review paper covers the following diseases: dementia, stroke, spinal cord injury, Parkinson's, and multiple sclerosis. The paper is organized as follows. Section Literature Review Method covers the literature review methodology in neurological disorders. Section Review of VR Studies in Neurological Diseases presents the results of the literature review and discusses the findings under the following three subsections: the effectiveness of VR in neurorehabilitation, Virtual Environments (VE), VR and interactivity devices, and intervention strategies and system evaluation. Section Emerging Technologies covers briefly the VR emerging technologies and the introduction of intelligent decision making and adaptive feedback in forthcoming VR applications. Finally, section Concluding Remarks provides some concluding remarks of the study.

LITERATURE REVIEW METHOD

The review was conducted based on Bargas-Avila and Hornbæk (2011) and Cochrane methodologies (Khan et al., 2001; Deeks et al., 2008), which consisted of the following five phases.

Procedure

Phase 1: Potentially Relevant Publications Identified

Electronic Libraries

We searched six electronic libraries, to cover a balanced range of disciplines, including computer science/engineering, medical research, and multidisciplinary sources. The libraries which included in the review were:

1. ACM Digital Library (ACM)
2. Google Scholar
3. IEEE Xplore (IEEE)
4. MEDLINE
5. PubMed
6. ScienceDirect (SD).

We restricted the search to a timeframe of five years (2014–2019), since we are aiming in only in fully immersive VR technologies have emerged for consumer use during this time (see examples given in **Table 1**).

Search terms

Our aim was to search for neurorehabilitation techniques that use immersive VR technology. Therefore, we have used the following two queries exactly to the aforementioned six libraries:

– Virtual Reality AND Neurorehabilitation
– Head Mounted Display AND Neurorehabilitation.

TABLE 1 | Selected VR technologies and indicative costing according to Amazon accessed September 2019.

VR technology	Release date	Cost	Company	Website
Google cardboard	25/06/2014	$5.71–$39.95	Google, US	https://vr.google.com/cardboard/get-cardboard/
Oculus gear VR	27/11/2015	$129.99	Oculus, US	https://www.oculus.com/gear-vr/
Oculus rift	28/03/2016	$399	Oculus, US	www.oculus.com/en-us/rift/
HTC vive	05/04/2016	$599–$1199	HTC, US	www.htcvive.com
Sony playstation	13/10/2016	$469.95–$549.95	Sony, AU	www.playstation.com/en-au/explore/playstation-vr/
Oculus GO	06/12/2016	$199–$249	Oculus, US	https://www.oculus.com/go/

FIGURE 1 | Selected VR HMDs from left to right, the Oculus GO, Oculus Quest, HTC VIVE wireless adapter, and PICO Neo.

TABLE 2 | Number of publications identified per library.

	ACM	Google scholar	IEEE	MEDLINE	PubMed	*SD*
Virtual reality AND neurorehabilitation	39	172	115	3	335	220
Head mounted display AND Neurorehabilitation	24	0	112	0	11	63
Total findings	1,094					

Search procedure

The above terms were searched in the following fields: full text (if available), title, abstract and keywords.

Search results

The total search that returned in phase 1 can be seen in **Table 2**. At the end of this phase, all corresponding PDFs were downloaded for the analysis to be conducted.

Phase 2: Publications Retrieved for Detailed Evaluation

First exclusion

A total of 1,069 articles were further analyzed after excluding manually 25 articles with wrong years entries.

Second exclusion

Duplicate publications across libraries (e.g., different libraries producing the same result) and *within* each library (e.g., different terms producing the same result within the same library) were removed.

We removed 32 duplicate publications across libraries, ending up with 1,047 different articles. After removing 36 duplicates *within* each library we ended up with 1,001 different articles.

Third exclusion

We narrowed the entries down to the original full articles that were written in English. We excluded 645 articles that we did not have access to the full length, 46 review articles, 37 articles that were not in English, and 18 articles that were not full peer-reviewed articles (e.g., referred to workshops, posters, presentations, magazine articles, theses). With these criteria, we excluded 746 articles. The remaining 255 articles comprised of journal and conference articles.

Phase 3: Publications to Be Included in the Analysis

Final exclusion

The focus on this review was placed on fully-immersive VR systems, therefore we excluded articles which used non-immersive or semi-immersive VR systems. Based on these criteria, we excluded 163 further articles which did not use fully-immersive VR technology and 8 articles that did not specify the type of VR equipment. We also excluded 24 articles that were not relevant to a nervous system injury linked to functional disability. Finally, we excluded 48 irrelevant articles that appeared in the first phase and were not excluded during the second phase filtering. These articles appeared in our search because they contain relevant words to the ones that we searched for, but did not match with the specific technology content. Based on these restrictions, in this phase we removed 240 irrelevant publications.

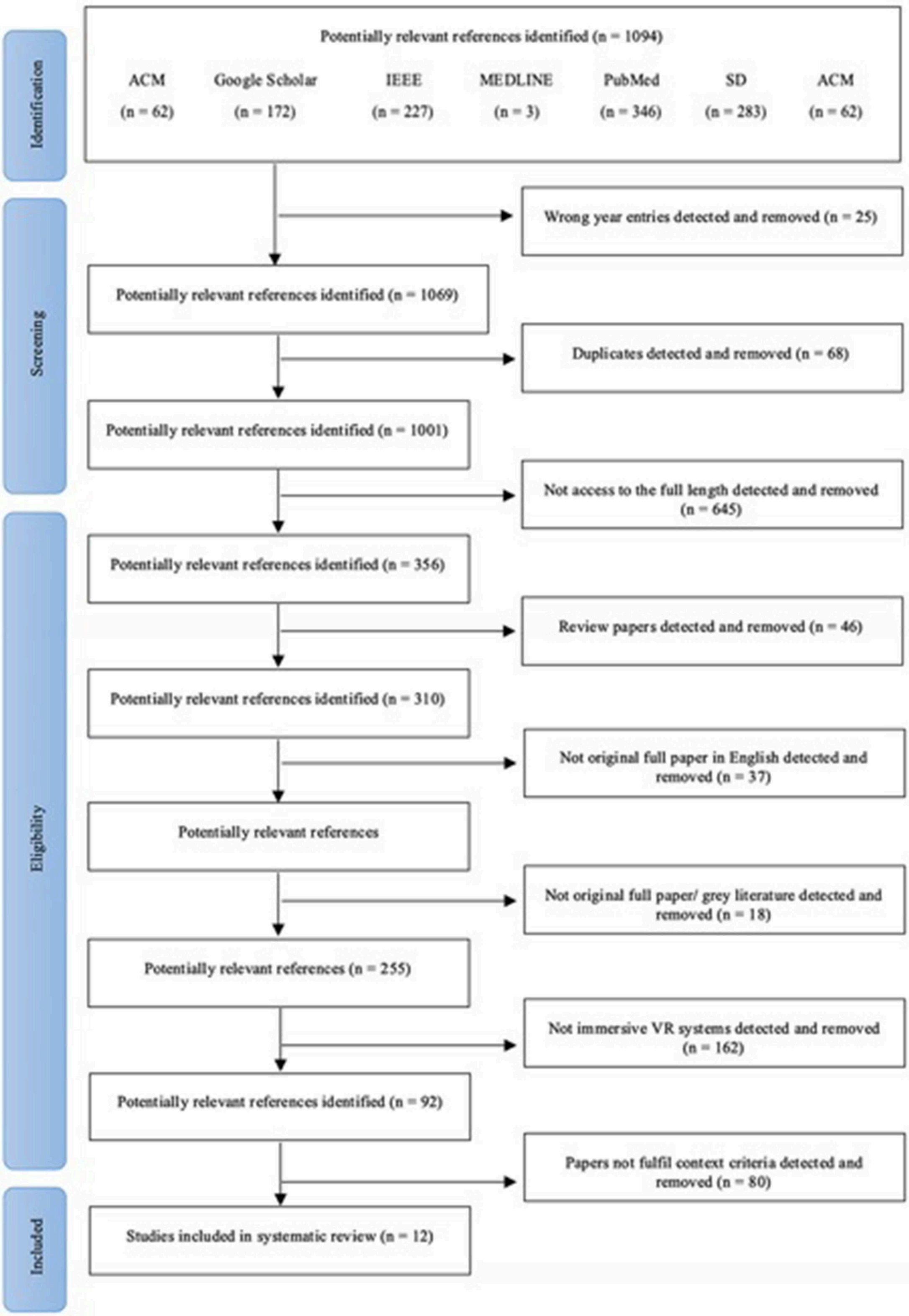

FIGURE 2 | Article identification and selection flow diagram.

As a result, we ended up with 12 relevant articles (10 journal articles and, 2 conference articles) (**Figure 2**).

Phase 4: Data Collection

In this phase, we extracted all the relevant information from the articles for the analysis to be conducted. Specifically, for each study, we recorded the objectives, the sample size, the condition or the population characteristics, the content of the VEs used, the interactivity devices used, the methodology/interventions the study was based on, other instruments used and the key findings. Moreover, we labeled each study, based on the result as positive (+), negative (-), or neutral ().

Phase 5: Data Analysis

Descriptive statistics were used to characterize the data from Phase 4. Thematic analysis was used as well to categorize our findings in themes, i.e., the population's characteristics, the types of the VEs, the interactivity devices used in the study, and the key findings. Inter-coder reliability was carried out to determine the correspondence of coding across researchers (between first and second author). Using the Cohen's Kappa formula, a reliability of 0.81 was computed.

TABLE 3 | Sample size/participants characteristics.

Study	Sample	Participant characteristics
Dementia		
Hodge et al., 2018	7	Dementia: 4 PwD; 3 Family Members
Mendez et al., 2015	5	Dementia
Rose et al., 2019	24	Dementia: 8 PwD; 16 Caregivers
Tabbaa et al., 2019	24	Dementia: 8 PwD; 16 Caregivers
Multiple Sclerosis		
Peruzzi et al., 2016	8	Multiple sclerosis
Parkinson		
Kim et al., 2017	33	Parkinson: 11 PD; 11 Healthy Young Adults; 11 Healthy Older Adults
Stroke		
Gamito et al., 2017	20	Stroke
Saleh et al., 2017	14	Stroke
Standen et al., 2017	27	Stroke: Arm dysfunction
Spinal Cord Injury		
Donati et al., 2016	8	Spinal cord injury
Pozeg et al., 2017	40	Spinal cord injury: 20 SCI; 20 Healthy—Control
Phantom Upper Limb Pain		
Ichinose et al., 2017	9	Phantom upper limb pain

REVIEW OF VR STUDIES IN NEUROLOGICAL DISEASES

All 12 studies examined the use of VR in samples with conditions of a nervous system injury linked to functional disability. In particular, most of the studies examined the use of VR for people living with dementia (PwD) ($n = 4$), stroke ($n = 3$), spinal cord injury ($n = 2$), parkinson's ($n = 1$), multiple sclerosis ($n = 1$), and phantom upper limb pain ($n = 1$). **Table 3** presents the sample size and the participant characteristics for each study.

The Effectiveness of Virtual Reality in Neuro-Rehabilitation

Overall, VR seems to show a promising potential for Neuro-Rehabilitation (**Table 4**). Ten out of 12 studies illustrated positive outcomes in the use of VR for the treatment of nervous system injury linked to functional disability. While the other two outlined the opportunities and challenges inherent to the design and use of VR with people with dementia and their careers (Hodge et al., 2018), and they used VR only as a tool to support the intervention for the treatment of stroke (Saleh et al., 2017).

Detailed analysis of the studies revealed that specific characteristics of the population, such as the type of disease, influence the study objectives, and the outcomes. With respect to the four studies of dementia, it was shown that all the studied objectives examined the feasibility of VR for people living with dementia (4/4). The feasibility of VR technology for people with dementia was examined with two different approaches. Two out of four studies (Hodge et al., 2018; Tabbaa et al., 2019) evaluated the technology feasibility from a patient-centered designed perspective targeting a human-computer interaction audience, whereas the rest of the studies adopted a psychology/psychiatric perspective to evaluate VR's feasibility (Mendez et al., 2015; Rose et al., 2019). All studies concluded that findings evidenced the clinical feasibility of VR for people with several stages of dementia. No adverse effects were stated, and high rates of pretense/immersion and positive emotional responses were reported.

Dementia was not the only disease that studies examined the feasibility of VR. From the review, it was found that multiple stroke (Standen et al., 2017), Parkinson (Kim et al., 2017), and sclerosis (Peruzzi et al., 2016) diseases were also linked to feasibility studies of VR. The results were in line with dementia studies. Importantly the VR's effectiveness was further enhanced by a study that examined the feasibility of long term (8 weeks) home-based VR of arm rehabilitation following stroke indicating that VR can be used as a personalized solution in home-based contexts (Standen et al., 2017).

VR was also used for neuropsychological rehabilitation based on a cognitive training program for stroke patients (Gamito et al., 2017). The results suggested that VR can be used as a cognitive training tool illustrating significant improvements in attention and memory functions. VR was also tested as a walk again rehabilitation tool for spinal cord injury patients. It demonstrated significant regain in voluntary motor control which resulted in walking improvements (Donati et al., 2016).

Finally, VR revealed promises in response to the treatment of phantom limb pain, since it was shown that tactile feedback via VR visual feedback was able to diminish pain and improve the analgesic effect of the affected limb (Ichinose et al., 2017).

Virtual Environments, Virtual Reality, and Interactivity Devices

The VR devices used for the treatment of nervous system injury linked to functional disabilities were eMagin Z800 (3/12),

TABLE 4 | VR effectiveness.

Study	Objectives	Results	Label
Dementia			
Hodge et al., 2018	(1) Design VR experiences for PwD; (2) Explore the reactions of PwD to VR; (3) Design a personalized experience.	Outline opportunities and challenges are inherent to the design and use of VR experiences with people with dementia and their careers.	()
Mendez et al., 2015	Assess the feasibility of VR and VR-Socialization for PwD.	(1) No adverse effects reported; (2) High rates of presence reported; (3) PwD tended to the greater verbal elaboration of answers in VR compared to real-world interviews.	(+)
Rose et al., 2019	Feasibility of VR for PwD.	(1) VR was tried and accepted by PwD; (2) PwD viewed VR as a 'change in the environment' and would use it again; (3) PwD experienced pleasure during and after VR and increased alertness because of VR; (4) Findings evidenced the clinical feasibility of VR for PwD.	(+)
Tabbaa et al., 2019	(1) Discuss the appeal and the impact of VR for PWD; (2) Present VR design opportunities, pitfalls, and recommendations for future deployment in healthcare services; (3) Demonstrate the potential of VR for PWD in locked settings.	VR is a feasible solution for PWD in long-term care.	(+)
Multiple Sclerosis			
Peruzzi et al., 2016	Assess the feasibility of VR treadmill for MS.	(1) Gait speed and stride length improved; (2) The ability to overcome obstacles was improved; (3) VR treadmill is feasible and safe for MS.	(+)
Parkinson			
Kim et al., 2017	Evaluate the safety of using VR for longer bouts of walking for individuals with PD.	(1) No adverse effects reported; (2) Lower Stress levels reported; (3) PD patients can successfully use VR during walking.	(+)
Stroke			
Gamito et al., 2017	Test the effectiveness of a VR for neuropsychological rehabilitation, a cognitive training program.	(1) Significant improvements in attention and memory functions; (2) The findings provide support for the use of VR cognitive training in neuropsychological rehabilitation.	(+)
Saleh et al., 2017	Test the interactions between regions in the brain that may be important for modulating the activation of the ipsilesional motor cortex during MVF.	Significant mirror feedback modulation of the ipsilesional motor cortex arising from the contralesional parietal cortex, in a region along the rostral extent of the intraparietal sulcus.	()
Standen et al., 2017	Feasibility of home-based VR of arm rehabilitation following stroke.	Significant improvement in the final Motor Activity Log.	(+)
Spinal Cord Injury			
Donati et al., 2016	Investigate the clinical impact of the Walk Again Rehabilitation, based on VR BMI.	(1) Neurological improvements in somatic sensation; (2) Regained voluntary motor control in key muscles; (3) Improvement in walking index; (4) 50% of patients upgraded to paraplegia classification.	(+)
Pozeg et al., 2017	Investigate changes in body ownership and chronic neuropathic pain in SCI using VR.	(1) SCI is less sensitive to multisensory stimulations inducing illusory leg ownership (2) Leg ownership decreased with time for SCI. (3) No differences between groups in global body ownership detected.	()
Phantom Upper Limb Pain			
Ichinose et al., 2017	Investigate the analgesic effect produced by tactile feedback using visual feedback.	(1) The pain was significantly lower during the VR Condition; (2) VR somatosensory feedback can improve the analgesic effect of the affected limb.	(+)

Google Cardboard (3/12), and Oculus Rift (3/12). The rest of the studies did not specify the VR equipment (3/12). Almost half of the studies (5/12) did not use any interactivity equipment and they used VR only to transport the patient into a different environment. Two studies used a Virtual Glove as interactivity device and the rest of the studies (5/12) used Xsens sensors, Vizard, Keyboard, EEG-based BMI, and Kinect to allow the user to interact with the VE.

From the analysis it was derived that most of the dementia studies used a Google Cardboard (3/4) (Hodge et al., 2018; Rose et al., 2019; Tabbaa et al., 2019) and an eMagin Z800 (1/4) (Mendez et al., 2015) VR device with no interactivity sensors

TABLE 5 | Virtual reality, interactivity devices and content.

Study	Virtual environments	VR device	Interactivity devices
Dementia			
Hodge et al., 2018	(1) A simple apartment, which allowed participants to turn their head and see out of a window; (2) A park, based on a local park in the area; (3) A tropical beach with a horse running along the sand.	Google cardboard	No
Mendez et al., 2015	The PwD was seated in a chair at the end of the conference table and told that they would be interviewed by the five avatars. They were asked to answer their questions as if they were real people. The avatars asked a series of questions.	eMagin Z800	No
Rose et al., 2019	(1) Cathedral; (2) Forest; (3) Sandy beach; (4) Rocky beach; (5) Countryside.	Google cardboard	No
Tabbaa et al., 2019	(1) Cathedral; (2) Forest; (3) Sandy beach; (4) Rocky beach; (5) Countryside.	Google cardboard	No
Multiple Sclerosis			
Peruzzi et al., 2016	A tree-lined trail with obstacles to appear on the trail.	eMagin Z800	Xsens
Parkinson			
Kim et al., 2017	A cityscape with buildings, animated avatars, and a straight sidewalk. Participants were able to freely look around the scene while walking.	Oculus rift	Vizard
Stroke			
Gamito et al., 2017	Several daily life activities: (1) Buy several items; (2) Find the way to the minimarket; (3) Find a virtual character dressed in yellow; (4) Recognize outdoor advertisements; (5) Digit retention.	eMagin Z800	Keyboard
Saleh et al., 2017	Hand mirror visual feedback in VR.	Not stated	Virtual glove
Standen et al., 2017	(1) Space-race: Pronation and supination of the hand to guide a spacecraft through obstacles; (2) Sponge-ball: Open their fist and extend their fingers to release a ball to hit a target. (3) Balloon-pop: Balloon was grasped and popped by moving it to a pin.	Not stated	Virtual glove
Spinal Cord Injury			
Donati et al., 2016	A 1st person's perspective virtual avatar body with rich visual and tactile feedback.	Oculus rift	EEG-based BMI
Pozeg et al., 2017	Virtual Avatar as a 3rd person perspective.	Not stated	No
Phantom Upper Limb Pain			
Ichinose et al., 2017	Repeatedly touched a target object with the affected limb, by converting via Mirror Visual Feedback the movements of the intact limb.	Oculus rift	Kinect

(4/4). Simple VEs with natural scenes were used by most of the studies (3/4). Based on these findings (**Table 5**) we can conclude that VR's feasibility for people with dementia does not require any expensive VR equipment and interactivity devices.

Patients with Parkinson (Kim et al., 2017) and multiple sclerosis (Peruzzi et al., 2016) were assigned to use Oculus Rift and eMagin Z800 VR devices paired with Xsens and Vizard sensors respectively. Both studies simulated walking VEs. A study with spinal cord injury patients (Donati et al., 2016) also used walking VEs based on EEG-based BMI interactivity device and an Oculus Rift HMD.

Two studies, with stroke (Saleh et al., 2017) and Phantom Limb pain Patients (Ichinose et al., 2017) used VR Oculus rift paired with Cyberlove and Kinect sensors, as an alternative solution to Mirror Box therapy. In mirror box therapy the patient was instructed to be seated in front of a mirror. The mirror's orientation was parallel to the patient's midline. At this position, the patient could see through the mirror the reflection of his/her unaffected body part. The affected body part was hidden beside the mirror and under the mirror box. This position created the visual illusion that the affected body part is working properly since visual cues were created through the mirror and from the opposite side of the unaffected body part in response to the brain's commands (Ramachandran, 2005). VR replicated the traditional mirror box in a technologically advanced version. More specifically, the mirror box was replaced by the VE and sensors to reproduce the movements of the unaffected body part. To conclude, the type of disease affects the selection of VEs, the VR and the interactivity devices.

Intervention Strategies and System Evaluation

The intervention strategies were divided in: (i) single testing, where the patient was exposed to the VR system only once, and (ii) multiple testings' where the patient used of the system for a long period of time incorporated into the rehabilitation training (i.e., from 6 weeks or up to a year) (**Table 6**).

In the aforementioned studies, dealing with people living with dementia, the feasibility of VR technology (3/4) was tested only once. Therefore, the intervention strategies were mostly associated with the development and the design of the technology from a patient-centered perspective (Hodge et al., 2018; Rose et al., 2019; Tabbaa et al., 2019). In particular, researchers along with clinical staff (Rose et al., 2019; Tabbaa et al., 2019)

TABLE 6 | Intervention strategies and system evaluation materials.

Study	Intervention Strategies	Evaluation Materials
Dementia		
Hodge et al., 2018	Single Intervention: VR Experiencing and co-design testing.	(1) Field notes; (2) Audio recordings; (3) Interviews.
Mendez et al., 2015	Single Intervention: PwD answered questions that were given by avatars.	(1) Interviews by VR avatars; (2) Heart Rate; (3) Self-reports: Arousal, Stress, Anxiety, Anger, Fatigue, Attention; (4) Interviews; (5) University of California at Los Angeles Structured Insight Interview; (6) Emotional Insight; (7) Mini-Mental State Examination; (8) Clinical Dementia Rating Scale; (9) Functional Activities Questionnaire; (10) Frontal Assessment Battery; (11) Frontal Systems Behavior Scale; (12) Wisconsin Card Sort Test.
Rose et al., 2019	Single Intervention: VR exposure as feasibility testing.	(1) Overt Aggression Scale-Modified for Neurorehabilitation; (2) St Andrews Sexual Behavior Assessment; (3) Observed Emotion Rating Scale; (4) Time; (5) Semi-structured Interviews.
Tabbaa et al., 2019	Single Intervention: VR exposure as feasibility testing.	(1) Overt Aggression Scale-Modified for Neurorehabilitation; (2) Observed Emotion Rating Scale; (3) Semi-structured Interviews (based on the System Usability Scale, Presence); (4) Observations.
Multiple Sclerosis		
Peruzzi et al., 2016	Six Weeks Training: Subjects were asked to walk over-ground in the gait analysis laboratory under two conditions: (a) at comfortable speed; (b) while serially subtracting the number "3" from a predefined 3-digit number.	Pre, Post, and Follow-up: (1) Collect Marker Trajectories and Ground Reaction Forces; (2) Joint kinematic Parameters (peak values of the kinematic curves); (3) Kinetic Parameters (maximum values of the joint moments and power during gait phases); (4) Six-minute Walk Test; (5) Square Step Test; (6) Expanded Disability Status Scale.
Parkinson		
Kim et al., 2017	Single Intervention: VR exposure of four bouts of 5 min walking to assess the feasibility of the VR walking.	(1) Movement Disorder Society Unified Parkinson's Disease Rating Scale; (2) Self-Selected Walking Speed; (3) Mini-Balance Evaluation Systems Test; (4) 14-item Balance Assessment for Dynamic Balance and Gait; (5) Activities-Specific Balance Confidence; (6) Center of pressure; (7) Simulator sickness questionnaire; (8) Stress Arousal Checklist; (9) Presence.
Stroke		
Gamito et al., 2017	Six Weeks Training: Randomly divided into 2 conditions: (1) VR 60 cognitive stimulation; (2) control waiting list.	(1) Wechsler Memory Scale; (2) Toulouse–Pieron Test; (3) Rey Complex Figure.
Saleh et al., 2017	Single Intervention: A VR goal-directed finger flexion movement with their unaffected hand while observing real-time visual feedback of the corresponding (veridical) or opposite (mirror) hand.	fMRI
Standen et al., 2017	Eight Weeks Training: Randomly divided into 2 conditions: (1) VR employing infrared capture to translate the position of the hand into gameplay or usual care; (2) Control - usual care.	(1) Wolf Motor Function Test; (2) Nine-Hole Peg Test; (3) Motor Activity Log; (4) Nottingham Extended Activities of Daily Living.
Spinal Cord Injury		
Donati et al., 2016	12 Months Training: (1) an immersive virtual reality environment in which a seated patient employed his/her brain activity, recorded via a 16-channel EEG, to control the movements of a human body avatar, while receiving visuotactile feedback; (2) identical interaction with the same virtual environment and BMI protocol while patients were upright, supported by a stand-in-table device; (3) training on a robotic body weight support (BWS) gait system on a treadmill; (4) training with a BWS gait system fixed on an overground track; (5) training with a brain-controlled robotic BWS gait system on a treadmill; (6) gait training with a brain-controlled, sensorized 12 degrees of freedom robotic exoskeleton. Clinical evaluation started on the first-day patients began training (Day 0) and was repeated after 4, 7, 10, and 12 months.	(1) American Spinal Injury Association; (2) Impairment Scale; (3) Semmes-Weinstein Monofilament Test; (4) Temperature Evaluation; (5) Lokomat L-force Evaluation; (6) Thoracic-Lumbar Scale; (7) Walking Index Spinal Cord Injury II; (8) Spinal Cord Independence Measurement III; (9) McGill Pain Questionnaire; (10) Visual Analog Scale; (11) Medical Research Council scale; (12) Modified Ashworth Scale; (13) Lokomat L-stiff Evaluation for spasticity; (14) World Health Organization Quality of Life Assessment Instrument-Bref; (15) Rosenberg Self-Esteem Scale; (16) Beck Depression Inventory.
Pozeg et al., 2017	Single Intervention: 2 × 2 repeated measures design, we manipulated the synchrony between the stroking of the virtual legs (synchronous/asynchronous) and the participant's back location (lower/upper back). In the synchronous condition, the stroking of the virtual legs was synchronized with the stroking of the participant's back. In the asynchronous condition, the visuotactile stimulation was delayed 1 s.	Questionnaires: (1) Body Illusions Studies; (2) Body ownership; (3) Visual Analog Scale; (4) Cambridge Depersonalization Scale.
Phantom Upper Limb Pain		
Ichinose et al., 2017	Single Intervention: Randomly divided in 3 conditions: (1) VR—applied tactile feedback to their cheek when their virtual affected limb touched a virtual object; (2) Control A—tactile feedback was either applied to their intact hand (Intact Hand Condition); (2) Control B—Not applied at all (No Stimulus Condition).	Pre and Post: McGill Pain Questionnaire.

and patients with dementia (Hodge et al., 2018) designed a VR system responsible to expose the patient into a different environment. All four studies used observation notes along with interview materials to evaluate the effectiveness of the system. Quantitative scales, such as arousal, stress, anxiety, anger, fatigue, and attention self-reports were also used to enhance the qualitative data (**Figure 3**).

The feasibility of VR was also tested for older adults with parkinson's enhanced by a walking task on a treadmill (Kim et al., 2017). Thirty-three participants (11 healthy young, 11 healthy older adults, and 11 individuals with PD) were recruited for this study and assigned to a 20 min walking tasks on a treadmill while watching a virtual city scene. Comparisons were made between the three different populations.

Patients with multiple sclerosis were asked to perform walking tasks on a treadmill watching a VR environment representing a tree-lined trail under a comfortable speed (Peruzzi et al., 2016). They were also asked to perform another walking task while serially subtracting the number "3" from a predefined 3-digit number. During the intervention, patients were required to pass obstacles aerating on the trail, while several dynamic distractors were also added to the VE to challenge the patient's attention. Each patient used a personalized environment based on personal gait problems (i.e., decreased foot clearance, obstacle avoidance, and problems with planning). Successful and unsuccessful passes, as determined by the inertial measurements, were rendered to the subject during the trial with visual and auditory feedback. A cognitive concurrent task was added by asking the subject to memorize the route to follow, which was shown to them prior to the trial. The training lasted for 6 weeks with each session to last about 45 min, with pre, post and follow-up materials to assess walking endurance and obstacle negotiation (**Figure 4**).

Apart from VR for walking tasks, interventions were also focused on affected upper limb training for patients dealing with stroke and phantom limb pain. In particular, Saleh et al. (2017) evaluated the effectiveness of VR with mirror visual feedback as a single intervention with the aim to facilitate recovery of disordered movement and stimulate activation of under-active brain areas due to stroke. During the experiment, patients were instructed to move the non-paretic hand's finger and watched the back-projected visual stimuli reflected in a mirror within the VR environment. The finger motion was back-projected onto a screen, showing two virtual hand models. On a given trial, the motion of the unaffected hand actuated one of the VR hands, located on the same (Veridical), or opposite (Mirror) side relative to the actual hand. The "move" prompt was displayed for the duration of the trial event (5s), and the "rest" prompt was displayed for the duration of the rest period (random 4–7-sec jittered). Subjects were instructed to complete the movement within the "move" epoch. Each scanning run included eight repetitions of four randomly interleaved visual feedback conditions and evaluated based on brain scanning reports. Similarly, mirror visual feedback was also used for phantom limb pain. Patients were instructed to touch via VR a virtual target. Once again during the experimental condition patients were instructed to move the non-affected hand to touch the virtual target and watched back in the VR the affected hand to perform the task. Pre and Post pain scales were used to evaluate the effectiveness of the system (**Figure 5**).

Finally, cognitive training intervention was also used via VR for the treatment of stroke (Gamito et al., 2017). The VR system was developed based on a serious games application for cognitive training, enhanced with attention and memory tasks consisting of daily life activities. The cognitive training VR scenarios were invented to train cognitive functions such as working memory tasks (i.e., buying several items), visuospatial orientation tasks (i.e., finding the way to the minimarket), and selective attention tasks (i.e., finding a virtual character dressed in yellow), recognition memory tasks (i.e., recognition of outdoor advertisements) and calculation (i.e., digit retention). Twenty stroke patients were randomly assigned to two conditions: exposure to the intervention and waiting list control to evaluate the effectiveness of using VR for cognitive training. Several scales were used to identify the effectiveness of the system (**Figure 6**).

EMERGING TECHNOLOGIES

Virtual Reality Input and Output Devices

New and emerging hardware developments are not yet commercially available. However, it is still possible to identify the technological trends particularly under the two main VR categories of input and output devices.

Input devices mostly refer to the controllers that are often enhanced by haptic feedback and hand and body tracking. A second input category is the navigation devices that bring to the user the illusion of moving through endless spaces within VEs such as one-direction and omnidirectional treadmills and passive low-friction surfaces, or "slidemills." Slidemill refers to devices like treadmills with the difference that the surface under the user's foot is static, therefore, the interface feels less natural and thus less immersive. Another form of input tracking system is hand and body tracking devices. User's posture estimation using inertial measurement units (IMUs) combined with magnetic tracking can be used to provide a reasonable self-representation in HMDs that elevates the feeling of realism in VEs. Finally, gesture tracking devices range from data gloves, with strain gauges or fiber optics that are often used combined with technologies using optical tracking and electromyography (EMG) signals that capture wrist movements with very promising prospects for VR applications in different fields especially for physical and cognitive rehabilitation.

Output devices primarily focus on the visual displays or more precisely wired or mobile HMDs when considering the VR field.

Wired HMDs specifications concentrate on quality factors like resolution, Field of View (FOV) or weight. Some wired HMDs are equipped with cameras for Augmented Reality (AR) applications and can be used as video see-through displays. Recently, the tendency in large VR companies is to include also eye tracking in the visual displays (e.g., Tobii VR[1], Steam FOVE[2], and SMI Eye tracking[3]).

[1]https://vr.tobii.com/
[2]https://www.getfove.com/
[3]https://www.smivision.com/

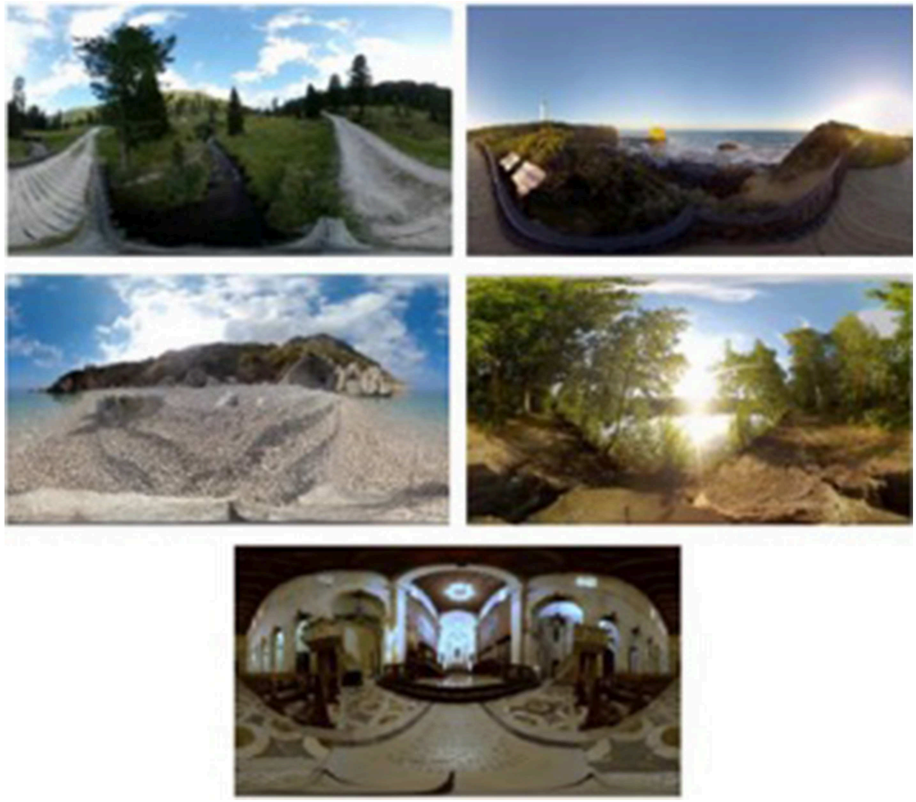

FIGURE 3 | Actual figure from Rose et al. (2019) paper, presenting the five options of VR environments given to patients with dementia.

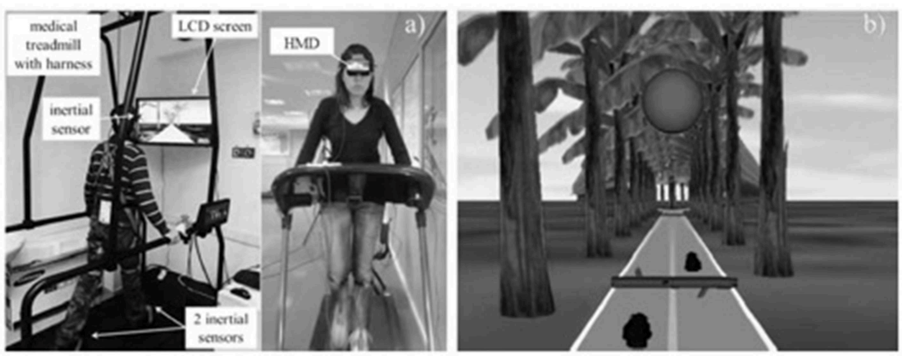

FIGURE 4 | Actual figure from Peruzzi et al. (2016) paper, presenting **(a)** The experimental set-up; **(b)** The virtual environment.

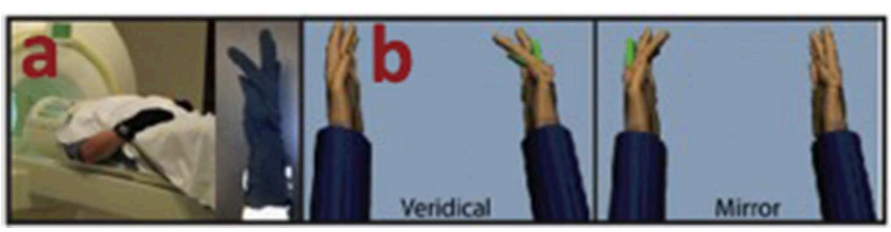

FIGURE 5 | Actual figure from Saleh et al. (2017) paper, presenting **(a)** the experimental set-up and equipment; **(b)** the virtual mirror feedback.

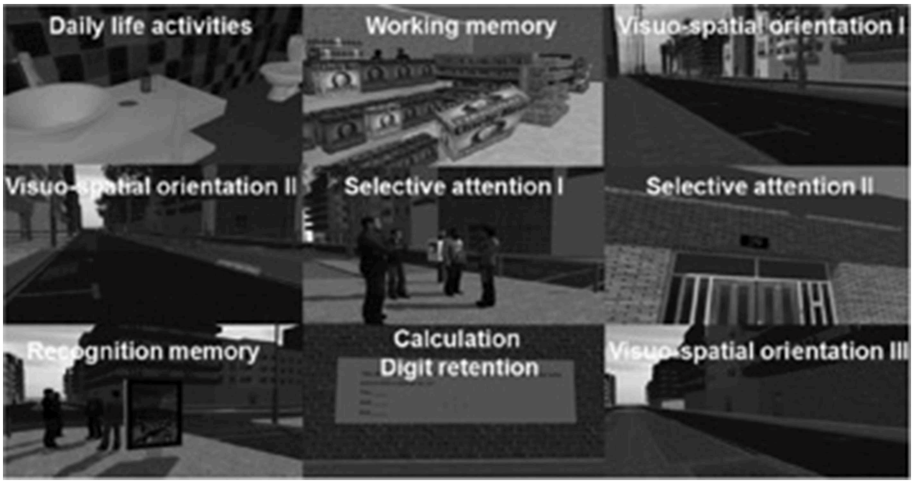

FIGURE 6 | Actual figure from Gamito et al. (2017) paper, presenting the nine virtual cognitive trainings.

On the other hand, mobile HMD systems run the applications wirelessly and without the need to be connected to a PC. Usually, these systems rely on smartphone technologies combined with ergonomically designed smartphone cases for stand-alone systems. Some examples of such standalone systems that have been released since 2018 include the Oculus Go[4], Oculus Quest[5], HTC VIVE focus[6], Pico Neo[7], and Xiaomi MI VR[8]. In addition to the later standalone systems, some manufacturers designed mobile devices with the option to use wireless adaptors for remote connection of the HMDs with PCs that run the VR applications (e.g., the HTC VIVE wireless adapter option[9]). Another important category of the output device are systems that include haptic and multi-sensory feedback. Haptic devices usually focus on a different sensory system with approaches that exist in the form of vests including Vibro-tactile elements. Ubiquitous displays providing sensory haptic feedback has also been undertaken like for instance, the example of viewing the effort to develop a system that generates airflow around the user to simulate weather conditions based on the application that the user is experiencing.

Other multisensory displays include head-mounted masks with the ability to produce different scents to further increase the feeling of immersion to the user as it was described by (Badler et al., 1992). Examples for multisensory devices involve integrated systems that blow cool and warm air in the users face or even combine ultrasonic ionizing systems that generate water mist (Matsukura et al., 2011). In addition, significant scientific research is being published with respect to olfactory information integrated into VR displays to increase the user's sense of presence in VR (Chen, 2006; Nakaizumi et al., 2006).

[4]https://www.oculus.com/go/
[5]https://www.oculus.com/quest/
[6]https://enterprise.vive.com/ca/vivefocus/
[7]https://www.pico-interactive.com/neo
[8]https://www.mi.com/global/mivr1c/
[9]https://www.vive.com/eu/wireless-adapter/

Intelligent Systems and Adaptive Feedback

Adaptation in a system involves a set of interacting entities that together can respond to changes and usually includes processing of feedback information from the output of the system to readjust the states of the system in a next time instance forming what is as "controlled close loops." Control loops in adaptive systems and machine learning are mostly used for prediction, recognition, detection, and optimization (Vaughan et al., 2016).

A recent literature review regarding the integration of computational intelligence and adaptation with VR technologies clearly demonstrated the prospects of achieving high impact results when combining these elements in application areas such as medicine, education training and gaming (Vaughan et al., 2016). Especially in applications that require trainee-specific and individual adaptive content, automation, machine learning and data driven features can guide feedback information to the inputs of autonomous systems and build new and customized training sessions based on individual requirements (Vaughan et al., 2016).

Some examples of self-adaptive systems in VR applications include automatically generated haptic, visual and auditory feedback signals that are used to modify the virtual scenarios and trigger methods to adapt the environmental behavior (e.g., Luzanin and Plancak, 2014). In addition, sensory information from assessment and scoring mechanisms, objectively facilitate the design of more optimum setups with automatically generating user-centered content (Wanzel et al., 2002; Vaughan et al., 2015).

Considering the above, adaptation and machine learning elements in rehabilitation tasks are very well suited because of the need to engage users and to intelligently adapt exercises based on user's progress (Borghese et al., 2013; Pirovano et al., 2013). In addition, adaptive feedback in rehabilitation tasks can supplement the therapist's input with the creation of a self-learned virtual therapist (Kallmann et al., 2015). For example, Borghese et al. (2013) presented an intelligent adaptive solution with Bayesian networks and fuzzy systems based on Nintendo

Kinect® motion sensing controllers for VR rehabilitation games (IGER) (Borghese et al., 2013).

Other examples include VR neurological rehabilitation systems that incorporate data mining of user scores and other measured performance data in a feedback computational intelligence loop to formulate a training plan for each trainee.

Future trends in virtual rehabilitation prescribe the path of new research for physiology driven adaptive VR systems this will allow the development of automated emotion recognition systems to be integrated in VR applications where the application responds appropriately to the emotions of their users (Popovic et al., 2009).

In addition, adaptive VR autonomous systems are currently enabling the performance of visio-haptic tasks without the requirement for human operator intervention. Accurate haptic simulation-based development platforms will inspire autonomous application with capabilities to convey the simulated VR information into a real-world haptic environment (like in surgery in autonomous neuro-rehabilitation tasks).

We consider that the technologies documented in this section will shape the development of the next generation of VR applications in rehabilitation. New virtual reality input devices will provide more complete data sets and signals about the behavior of the patient demanding intelligent processing, monitoring and profiling of the patient toward offering a personalized VR rehabilitation solution. Similarly, new output devices will facilitate VR applications to be more realistic, personalized and closer to the rehabilitation needs of the patient.

The aforementioned technologies will shape the development of state of the art VR rehabilitation services in the framework of emerging connected health systems and services (Pattichis and Panayides, 2019) in support of 4P's medicine (Golubnitschaja et al., 2016). More specifically, emerging VR applications will be (Golubnitschaja et al., 2016): (i) predictive: VR systems will automatically capture data to predict, manage, adapt, and/or deliver better treatment plans; (ii) pre-emptive: VR solutions will be designed to monitor vital signs and activities in real time which will communicate with personal health record archives and healthcare professionals; (iii) personalized interventions: new VR applications will enable the offering of best possible, most optimal, and innovative treatments; (iv) participatory: patient-centric VR applications will empower patients to be more active and allow the sharing of experiences. It is expected that emerging VR applications sharing the 4P's concept will trigger the offering of new services and business models for the benefit of the citizen.

CONCLUDING REMARKS

Recent advances in VR immersive technologies provide new methods and tools for the development of novel and promising applications mainly for neurological rehabilitation. VR interventions have several advantages and are rapidly gaining ground as popular applications for different disease conditions. The big advantage of VR applications in rehabilitation is that they offer a "real-life like environment." In addition, VR applications advantages include, control of stimulus presentation, and response measurements, safe assessment on different unsafe rehabilitation tasks, easy learning of the tasks to be performed, standardization of rehabilitation protocols, and enhanced user interaction and empowerment.

On the other hand, limitations of VR interventions include that the patient might forget that he/she is in a testing situation and the difficulty and complexity in generating personalized training environments. These prescribe some of the existing challenges to develop low-cost rehabilitation assessment and monitoring environments and applications. Furthermore, the development of VR technologies in recent years have resulted in more accessible and less expensive solutions, which could still provide positive results. However, the full potential of VR applications in healthcare still remains to be explored.

The purpose of this research work was to carry out a systematic review of emerging VR applications developed over the last 5 years, covering certain neurological diseases. Although, the final number of studies analyzed is rather small (12), still valuable input can be gained. It is expected, that the number of studies in consumer-oriented fully-immersive VR systems will significantly increase in the near future, given the rapid progression of development both in the hardware and software in these technologies.

The findings of this systematic literature review showed positive and promising results of using VR for rehabilitation exercise. It also suggests that low-cost, immersive VR technologies can prove to be effective for clinical rehabilitation in healthcare and home-based settings with practical implications and uses. Based on our review we found that dementia studies used the cheapest VR equipment (Goggles Cardboard) and no interactivity devices, achieving very good results. In addition, low-cost VR devices were found to be free of adverse effects, and high rates of presence/immersion, and positive emotional responses were reported. Consequently, it is now conceivable to use VR low-cost technologies with no interactivity devices to expose people with dementia in different environments, to improve pleasure and alertness. The application can evolve based on the needs and available budget one can have. It is also possible to experience VR outside of a specialized laboratory, making it more accessible to a wider group of patients if needed.

Even though most dementia studies used low-cost VR equipment with no interactivity devices, the rest of the studies apart from one (Spinal Cord Injury—Pozeg et al., 2017) used the following VR systems: Xsens, Vizard, EEG-based BM, Cyberlove, and Kinect sensors. These interactivity devices were responsible to transport the patients' movement into the VR environment in order to enhance the physical or cognitive training. VR and interactivity devices resulted in the development of a holistic portable, accessible and usable systems enabling the better handling of the neurological disorders reported. Furthermore, by employing machine learning and AI in VR applications, exercise interventions can be patient's specific to the treatment needs of the patient, thus, offering optimal care. Complex virtual therapy exercises need to be created with precise control over the stimulus and cognitive capacity that the user will experience.

Concluding, the main findings of this systematic literature review indicated that VR technology could be effective in improving the condition of the patient for certain neurological diseases. This review study outlined some key factors that may contribute to the effectiveness of VR applications, such as the objective of the study linked with the intervention strategy, the VR technology and interactivity equipment used in the study and other. It is expected that VR applications in healthcare will flourish within the next few years, triggering further investigations in different clinical settings. It is hoped that these VR applications could also prove to have an impact on the wellness of the patient that remains to be thoroughly investigated.

AUTHOR CONTRIBUTIONS

ES and MM conceived of the original idea. KN and CP supervised and assisted the findings of this work. All authors discussed the results and contributed to the final manuscript.

ACKNOWLEDGMENTS

This research is part of the project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 739578 (RISE—Call: H2020-WIDESPREAD-01-2016-2017-TeamingPhase2) and the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

REFERENCES

Anthes, C., García-Hernández, R. J., Wiedemann, M., and Kranzlmüller, D. (2016). "State of the art of virtual reality technology," in *2016 IEEE Aerospace Conference* (Big Sky, MT: IEEE), 1–19. doi: 10.1109/AERO.2016.7500674

Badler, N., Phillips, C., and Webber, B. (1992). *Simulating Humans, Computer Graphics, and Control*. New York, NY: Oxford University Press.

Bargas-Avila, J. A., and Hornbæk, K. (2011). "Old wine in new bottles or novel challenges: a critical analysis of empirical studies of user experience," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC: ACM), 2689–2698. doi: 10.1145/1978942.1979336

Borghese, N. A., Pirovano, M., Lanzi, P. L., Wüest, S., and de Bruin, E. D. (2013). Computational intelligence and game design for effective at-home stroke rehabilitation. *Games Health* 2, 81–88. doi: 10.1089/g4h.2012.0073

Chen, Y. (2006). "Olfactory display: development and application in virtual reality therapy," in *16th International Conference on Artificial Reality and Telexistence–Workshops (ICAT'06)* (Hangzhou: IEEE), 580–584. doi: 10.1109/ICAT.2006.95

Deeks, J. J., Higgins, J. P., and Altman, D. G. (2008). "Analysing data and undertaking meta-analyses," in *Cochrane Handbook for Systematic Reviews of Interventions*, eds J. P. Higgins and S. Green (Oxford: The Cochrane Collaboration), 243–296. doi: 10.1002/9780470712184.ch9

Donati, A. R., Shokur, S., Morya, E., Campos, D. S., Moioli, R. C., Gitti, C. M., et al. (2016). Long-term training with a brain-machine interface-based gait protocol induces partial neurological recovery in paraplegic patients. *Sci. Rep*. 6:30383. doi: 10.1038/srep30383

Gamito, P., Oliveira, J., Coelho, C., Morais, D., Lopes, P., Pacheco, J., et al. (2017). Cognitive training on stroke patients via virtual reality-based serious games. *Disabil. Rehabil.* 39, 385–388. doi: 10.3109/09638288.2014.934925

Golubnitschaja, O., Baban, B., Boniolo, G., Wang, W., Bubnov, R., Kapalla, M., et al. (2016). Medicine in the early twenty-first century: paradigm and anticipation-EPMA position paper 2016. *EPMA J.* 7, 23–36. doi: 10.1186/s13167-016-0072-4

Hodge, J., Balaam, M., Hastings, S., and Morrissey, K. (2018). "Exploring the design of tailored virtual reality experiences for people with dementia," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal, QC: ACM), 514. doi: 10.1145/3173574.3174088

Ichinose, A., Sano, Y., Osumi, M., Sumitani, M., Kumagaya, S. I., and Kuniyoshi, Y. (2017). Somatosensory feedback to the cheek during virtual visual feedback therapy enhances pain alleviation for phantom arms. *Neurorehabil. Neural Repair* 31, 717–725. doi: 10.1177/1545968317718268

Kallmann M., Camporesi C., and Han J. (2015). "VR-assisted physical rehabilitation: adapting to the needs of therapists and patients," in *Virtual Realities. Lecture Notes in Computer Science*, Vol. 8844, eds, G. Brunnett, S. Coquillart, R. van Liere, G. Welch, and L. Váša (Cham: Springer), 147–168. doi: 10.1007/978-3-319-17043-5_9

Khan, K. S., Ter Riet, G., Glanville, J., Sowden, A. J., and Kleijnen, J. (2001). *Undertaking systematic reviews of research on effectiveness: CRD's guidance for carrying out or commissioning reviews (No. 4 (2n)*. NHS Centre for Reviews and Dissemination.

Kim, A., Darakjian, N., and Finley, J. M. (2017). Walking in fully immersive virtual environments: an evaluation of potential adverse effects in older adults and individuals with Parkinson's disease. *J. Neuroeng. Rehabil.* 14:16. doi: 10.1186/s12984-017-0225-2

Luzanin, O., and Plancak, M. (2014). Hand gesture recognition using low-budget data glove and cluster-trained probabilistic neural network. *Assembly Automat.* 34, 94–105. doi: 10.1108/AA-03-2013-020

Ma, M., and Zheng, H. (2011). "Virtual reality and serious games in healthcare," in *Advanced Computational Intelligence Paradigms in Healthcare 6. Virtual Reality in Psychotherapy, Rehabilitation, and Assessment* (Berlin Heidelberg: Springer), 169–192. doi: 10.1007/978-3-642-17824-5_9

Matsangidou, M., Ang, C. S., and Sakel, M. (2017). Clinical utility of virtual reality in pain management: a comprehensive research review. *Br. J. Neurosci. Nurs.* 13, 133–143. doi: 10.12968/bjnn.2017.13.3.133

Matsukura, H., Nihei, T., and Ishida, H. (2011). "Multi-sensorial field display: presenting spatial distribution of airflow and odor," in *2011 IEEE Virtual Reality Conference* (Singapore: IEEE), 119–122. doi: 10.1109/VR.2011.5759448

Mendez, M. F., Joshi, A., and Jimenez, E. (2015). Virtual reality for the assessment of frontotemporal dementia, a feasibility study. *Disabil. Rehabil.* 10, 160–164. doi: 10.3109/17483107.2014.889230

Nakaizumi, F., Noma, H., Hosaka, K., and Yanagida, Y. (2006). "SpotScents: a novel method of natural scent delivery using multiple scent projectors," in *IEEE Virtual Reality Conference (VR 2006)* (Alexandria, VA: IEEE), 207–214. doi: 10.1109/VR.2006.122

Pattichis, C. S., and Panayides, A. (2019). *Specialty grand challenge article, Frontiers in Digital* Health, *Connected Health Specialty Section*. doi: 10.3389/fdgth.2019.00001

Peruzzi, A., Cereatti, A., Della Croce, U., and Mirelman, A. (2016). Effects of a virtual reality and treadmill training on gait of subjects with multiple sclerosis: a pilot study. *Mult. Scler. Relat. Disord.* 5, 91–96. doi: 10.1016/j.msard.2015.11.002

Pirovano, M., Lanzi, P. L., Mainetti, R., and Borghese, N. A. (2013). "IGER: a game engine specifically tailored to rehabilitation," in *Games for Health* (Wiesbaden: Springer Vieweg), 85–98. doi: 10.1007/978-3-658-02897-8_7

Popovic, S., Horvat, M., Kukolja, D., Dropuljic, B., and Cosic, K. (2009). Stress inoculation training supported by physiology-driven adaptive virtual reality stimulation. *Stud. Health Technol. Inform.* 144, 50–54. doi: 10.3233/978-1-60750-017-9-50

Pozeg, P., Palluel, E., Ronchi, R., Solcà, M., Al-Khodairy, A. W., Jordan, X., et al. (2017). Virtual reality improves embodiment and neuropathic pain caused by spinal cord injury. *Neurology* 18, 1101–1212. doi: 10.1212/WNL.0000000000004585

Ramachandran, V. S. (2005). Plasticity and functional recovery in neurology. *Clin. Med.* 5, 368–373. doi: 10.7861/clinmedicine.5-4-368

Rose, V., Stewart, I., Jenkins, K., Tabbaa, L., Ang, C. S., and Matsangidou, M. (2019). *Exploring the Feasibility of Virtual Reality Distraction With Individuals Living With Dementia*. Dementia: The International Journal of Social Research and Practice.

Saleh, S., Yarossi, M., Manuweera, T., Adamovich, S., and Tunik, E. (2017). Network interactions underlying mirror feedback in stroke: a dynamic causal modeling study. *NeuroImage Clin.* 13, 46–54. doi: 10.1016/j.nicl.2016.11.012

Standen, P. J., Threapleton, K., Richardson, A., Connell, L., Brown, D. J., Battersby, S., et al. (2017). A low cost virtual reality system for home based rehabilitation of the arm following stroke: a randomised controlled feasibility trial. *Clin. Rehabil.* 31, 340–350. doi: 10.1177/0269215516640320

Tabbaa, L., Ang, C. S., Rose, V., Panote, S., Stewart, I., Jenkins, K., et al. (2019). "Bringing the outside in: VR experiences for individuals with dementia in a locked psychiatric hospital," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow: ACM), 514. doi: 10.1145/3290605.3300466

Vaughan, N., Dubey, V. N., Wainwright, T. W., and Middleton, R. G. (2015). "Can virtual-reality simulators assess experience and skill level of orthopaedic surgeons?" in *2015 Science and Information Conference (SAI)* (IEEE), 105–108. doi: 10.1109/SAI.2015.7237133

Vaughan, N., Gabrys, B., and Dubey, V. N. (2016). An overview of self-adaptive technologies within virtual reality training. *Comp. Sci. Rev.* 22, 65–87. doi: 10.1016/j.cosrev.2016.09.001

Wanzel, K. R., Ward, M., and Reznick, R. K. (2002). Teaching the surgical craft: from selection to certification. *Curr. Probl. Surg.* 39, 583–659. doi: 10.1067/mog.2002.123481

Weber, H. (2019). *Virtual Reality in Healthcare - Top 5 Uses*. Retrieved from: https://www.neurorehabvr.com/blog/2019/1/29virtual-reality-in-healthcare

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Robotics and AI

REVIEW
published: 23 June 2020
doi: 10.3389/frobt.2020.00071

Elderly Fall Detection Systems: A Literature Survey

Xueyi Wang[1]*, *Joshua Ellul*[2] *and George Azzopardi*[1]

[1] Department of Computer Science, Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Groningen, Netherlands, [2] Computer Science, Faculty of Information & Communication Technology, University of Malta, Msida, Malta

Falling is among the most damaging event elderly people may experience. With the ever-growing aging population, there is an urgent need for the development of fall detection systems. Thanks to the rapid development of sensor networks and the Internet of Things (IoT), human-computer interaction using sensor fusion has been regarded as an effective method to address the problem of fall detection. In this paper, we provide a literature survey of work conducted on elderly fall detection using sensor networks and IoT. Although there are various existing studies which focus on the fall detection with individual sensors, such as wearable ones and depth cameras, the performance of these systems are still not satisfying as they suffer mostly from high false alarms. Literature shows that fusing the signals of different sensors could result in higher accuracy and lower false alarms, while improving the robustness of such systems. We approach this survey from different perspectives, including data collection, data transmission, sensor fusion, data analysis, security, and privacy. We also review the benchmark data sets available that have been used to quantify the performance of the proposed methods. The survey is meant to provide researchers in the field of elderly fall detection using sensor networks with a summary of progress achieved up to date and to identify areas where further effort would be beneficial.

Keywords: fall detection, Internet of Things (IoT), information system, wearable device, ambient device, sensor fusion

OPEN ACCESS

Edited by:
Soumik Sarkar,
Iowa State University, United States

Reviewed by:
Sambuddha Ghosal,
Massachusetts Institute of Technology, United States
Carl K. Chang,
Iowa State University, United States

Correspondence:
Xueyi Wang
xueyi.wang@rug.nl

Specialty section:
This article was submitted to Sensor Fusion and Machine Perception, a section of the journal Frontiers in Robotics and AI

Received: *17 December 2019*
Accepted: *30 April 2020*
Published: *23 June 2020*

Citation:
Wang X, Ellul J and Azzopardi G (2020) Elderly Fall Detection Systems: A Literature Survey. Front. Robot. AI 7:71. doi: 10.3389/frobt.2020.00071

1. INTRODUCTION

More than nine percent of the population of China was aged 65 or older in 2015 and within 20 years (2017–2037) it is expected to reach 20%[1]. According to the World Health Organization (WHO), around 646 k fatal falls occur each year in the world, the majority of whom are suffered by adults older than 65 years (WHO, 2018). This makes it the second reason for unintentional injury death, followed by road traffic injuries. Globally, falls are a major public health problem for the elderly. Needless to say, the injuries caused by falls that elderly people experience have many consequences to their families, but also to the healthcare systems and to the society at large.

As illustrated in **Figure 1**, Google Trends[2] show that fall detection has drawn increasing attention from both academia and industry, especially in the last couple of years, where a sudden increase can be observed. Moreover, on the same line, the topic of fall-likelihood prediction is very significant too, which is coupled with some applications focused on prevention and protection.

[1] https://chinapower.csis.org/aging-problem/
[2] https://www.google.com/trends

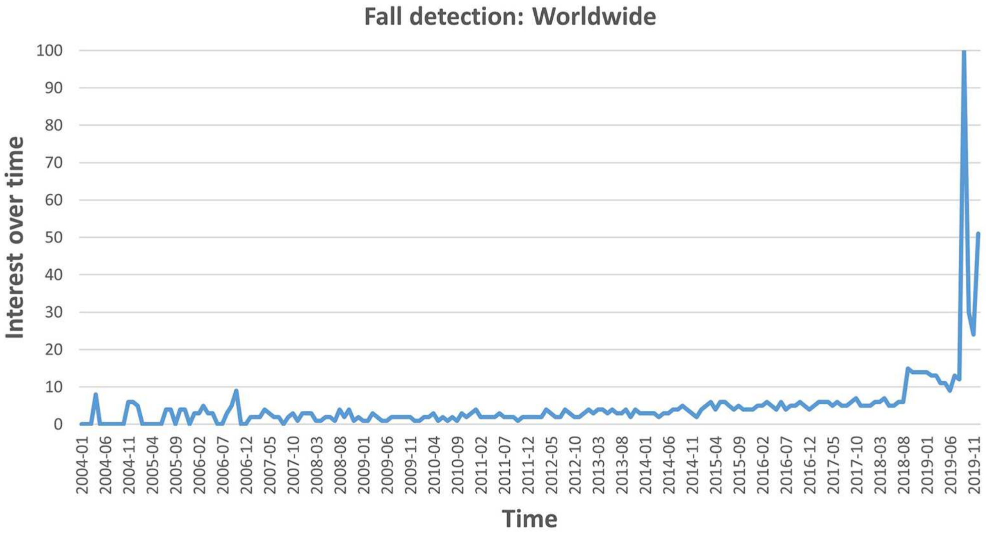

FIGURE 1 | Interest of fall detection over time, from January 2004 to December 2019. The data is taken from Google Trends with the search topic "fall detection." The values are normalized with the maximum interest, such that the highest interest has a value of 100.

El-Bendary et al. (2013) reviewed the trends and challenges of elderly fall detection and prediction. Detection techniques are concerned with recognizing falls *after* they occur and trigger an alarm to emergency caregivers, while predictive methods aim to forecast fall incidents *before* or *during* their occurrence, and therefore allow immediate actions, such as the activation of airbags.

During the past decades, much effort has been put into these fields to improve the accuracy of fall detection and prediction systems as well as to decrease the false alarms. **Figure 2** shows the top 25 countries in terms of the number of publications about fall detection from the year 1945 to 2020. Most of the publications originate from the United States, followed by England, China, and Germany, among others. The data indicates that developed countries invest more in conducting research in this field than others. Due to higher living standards and better medical resources, people in developed countries are more likely to have longer life expectancy, which results in a higher aging population in such countries (Bloom et al., 2011).

In this survey paper, we provide a holistic overview of fall detection systems, which is aimed for a broad readership to become abreast with the literature in this field. Besides fall detection modeling techniques, this review covers other topics including issues pertaining to data transmission, data storage and analysis, and security and privacy, which are equally important in the development and deployment of such systems.

The other parts of the paper are organized as follows. In section 2, we start by introducing the types of fall and reviewing other survey papers to illustrate the research trend and challenges up to date, followed by a description of our literature search strategy. Next, in section 3 we introduce hardware and software components typically used in fall detection systems. Sections 4 and 5 give an overview of fall detection methods that rely on both individual or a collection of sensors. In section 6, we address issues of security and privacy. Section 7 introduces projects and applications of fall detection. In section 8, we provide a discussion about the current trends and challenges, followed by a discussion on challenges, open issues, and other aspects on future directions. Finally, we provide a summary of the survey and draw conclusions in section 9.

2. TYPES OF FALLS AND PREVIOUS REVIEWS ON ELDERLY FALL DETECTION

2.1. Types of Falls

The impact and consequences of a fall can vary drastically depending upon various factors. For instance, falling whilst either walking, standing, sleeping or sitting on a chair, share some characteristics in common but also have significant differences between them.

In El-Bendary et al. (2013), the authors group the types of falls in three basic categories, namely *forward*, *lateral*, and *backward*. Putra et al. (2017) divided falls into a broader set of categories, namely *forward*, *backward*, *left-side*, *right-side*, *blinded-forward*, and *blinded-backward*, and in the study by Chen et al. (2018) falls are grouped in more specific categories including *fall lateral left lie on the floor*, *fall lateral left and sit up from floor*, *fall lateral right and lie on the floor*, *fall lateral and left sit up from the floor*, *fall forward and lie on the floor*, and *fall backward and lie on the floor*.

Besides the direction one takes whilst falling another important aspect is the duration of the fall, which may be influenced by age, health and physical condition, along with any consequences of activities that the individual was undertaking. Elderly people may suffer from longer duration of falls, because of motion with low speed in the activity of daily living. For instance, in fainting or chest pain related episodes an elderly person might try to rest by a wall before lying on the floor. In other situations, such as injuries due to obstacles or dangerous

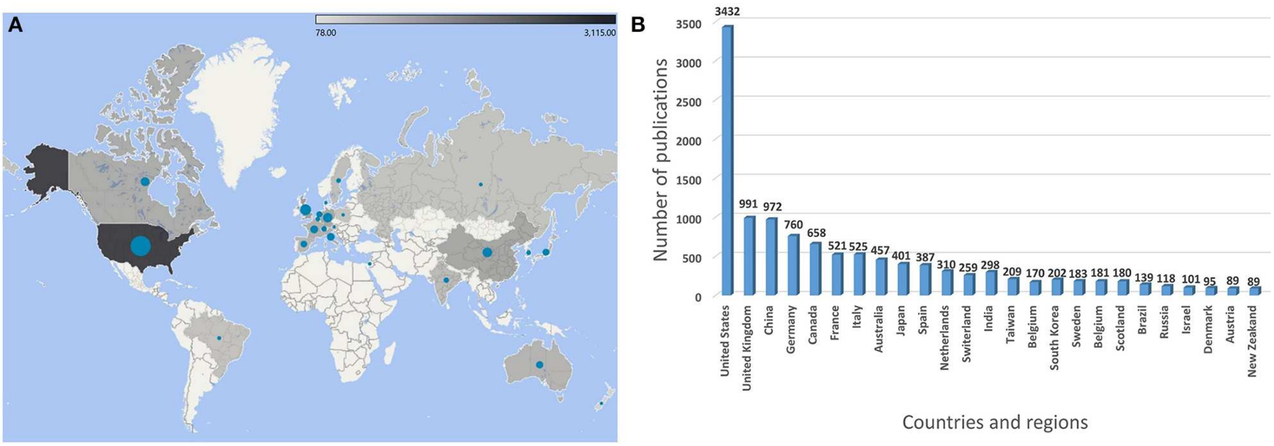

FIGURE 2 | (A) A map and **(B)** a histogram of publications on fall detection by countries and regions from 1945 to 2020.

settings (e.g., slanting or uneven pavement or surfaces), an elderly person might fall abruptly. The age and gender of the subject also play a role in the kinematics of falls.

The characteristics of different types of falls are not taken into consideration in most of the work on fall detection surveyed. In most of the papers to date, data sets typically contain falls that are simulated by young and healthy volunteers and do not cover all types of falls mentioned above. The resulting models from such studies, therefore, do not lead to models that generalize well enough in practical settings.

2.2. Review of Previous Survey Papers

There are various review papers that give an account of the development of fall detection from different aspects. Due to the rapid development of smart sensors and related analytical approaches, it is necessary to re-illustrate the trends and development frequently. We choose the most highly cited review papers, from 2014 to 2020, based on Google Scholar and Web of Science, and discuss them below. These selected review papers demonstrate the trends, challenges, and development in this field. Other significant review papers before 2014 are also covered in order to give sufficient background of earlier work.

Chaudhuri et al. (2014) conducted a systematic review of fall detection devices for people of different ages (excluding children) from several perspectives, including background, objectives, data sources, eligibility criteria, and intervention methods. More than 100 papers were selected and reviewed. The selected papers were divided into several groups based on different criteria, such as the age of subjects, method of evaluation and devices used in detection systems. They noted that most of the studies were based on synthetic data. Although simulated data may share common features with real falls, a system trained on such data cannot reach the same reliability of those that use real data.

In another survey, Zhang et al. (2015) focused on vision-based fall detection systems and their related benchmark data sets, which have not been discussed in other reviews. Vision-based approaches of fall detection were divided into four categories, namely individual single RGB cameras, infrared cameras, depth cameras, and 3D-based methods using camera arrays. Since the advent of depth cameras, such as Microsoft Kinect, fall detection with RGB-D cameras has been extensively and thoroughly studied due to the inexpensive price and easy installation. Systems which use calibrated camera arrays also saw prominent uptake. Because such systems rely on many cameras positioned at different viewpoints, challenges related to occlusion are typically reduced substantially, and therefore result in less false alarm rates. Depth cameras have gained particular popularity because unlike RGB camera arrays they do not require complicated calibration and they are also less intrusive of privacy. Zhang et al. (2015) also reviewed different types of fall detection methods, that rely on the activity/inactivity of the subjects, shape (width-to-height ratio), and motion. While that review gives a thorough overview of vision-based systems, it lacks an account of other fall detection systems that rely on non-vision sensors such as wearable and ambient ones.

Further to the particular interest in depth cameras, Cai et al. (2017) reviewed the benchmark data sets acquired by Microsoft Kinect and similar cameras. They reviewed 46 public RGB-D data sets, 20 of which are highly used and cited. They compared and highlighted the characteristics of all data sets in terms of their suitability to certain applications. Therefore, the paper is beneficial for scientists who are looking for benchmark data sets for the evaluation of new methods or new applications.

Based on the review provided by Chen et al. (2017a), individual depth cameras and inertial sensors seem to be the most significant approaches in vision- and non-vision-based systems, respectively. In their review, the authors concluded that fusion of both types of sensor resulted in a system that is more robust than a system relying on one type of sensor.

The ongoing and fast development in electronics have resulted in more miniature and cheaper electronics. For instance, the survey by Igual et al. (2013) noted that low-cost cameras and accelerometers embedded in smartphones may offer the most sensible technological choice for the investigation of fall detection. Igual et al. (2013) identified two main trends on how research is progressing in this field, namely the use of vision and smartphone-based sensors that give input and the use of machine learning for the data analysis. Moreover, they reported the following three main challenges: (i) real-world deployment performance, (ii) usability, and (iii) acceptance. Usability refers to how practical the elderly people find the given system. Because of the issue of privacy and intrusive characteristics of some sensors, there is a lack of acceptance for the elderly to live in an environment monitored by sensors. They also pointed out several issues which need to be taken into account, such as smartphone limitations (e.g., people may not carry smartphones all the time with them), privacy concerns, and the lack of benchmark data sets of realistic falls.

The survey papers mentioned above focus mostly on the different types of sensors that can be used for fall detection. To the best of our knowledge, there are no literature surveys that provide a holistic review of fall detection systems in terms of data acquisition, data analysis, data transport and storage, sensor networks and Internet of Things (IoT) platforms, as well as security and privacy, which are significant in the deployment of such systems.

2.3. Key Results of Pioneering Papers

In order to illustrate a timeline of fall detection development, in this section we focus on the key and pioneering papers. Through manual filtering of papers using the web of science, one can find the trendsetting and highly cited papers in this field. By analyzing retrieved articles using citespace one can find that fall detection research first appeared in the 1990s, beginning with the work by Lord and Colvin (1991) and Williams et al. (1998). A miniature accelerometer and microcomputer chip embedded in a badge was used to detect falls (Lord and Colvin, 1991), while Williams et al. (1998) applied a piezoelectric shock sensor and a mercury tilt switch which monitored the orientation of the body to detect falls. At first, most studies were based on accelerometers including the work by Bourke et al. (2007). In their work, they compared which of the trunk and thigh offer the best location to attach the sensor. Their results showed that a person's trunk is a better location in comparison to the thigh, and they achieved 100% specificity with a certain threshold value with a sensor located in the trunk. This method was the state-of-the-art at the time, which undoubtedly supported it in becoming the most highly cited paper in the field.

At the time the trend was to use individual sensors for detection, within which another key paper by Bourke and Lyons (2008) was proposed to explore the problem at hand by using a single gyroscope that measures three variables, namely angular velocity, angular acceleration, and the change in the subject's trunk-angle. If the values of these three variables in a particular instance are above some empirically determined thresholds, then that instance is flagged as a fall. Three thresholds were set to distinguish falls from non-falls. Falls are detected when the angular velocity of a subject is greater than the fall threshold, and the angular acceleration of the subject is greater than the second fall threshold, and the change in the trunk-angle of the subject is greater than the third fall threshold. They reported accuracy of 100% on a data set with only four kinds of falls and 480 movements simulated by young volunteers. However, for those classifiers, which are based solely on either accelerometers or gyroscopes, are argued to suffer from insufficient robustness (Tsinganos and Skodras, 2018). Later, Li et al. (2009) investigated fusion of gyroscope and accelerometer data for the classification of falls and non-falls. In their work, they demonstrated how a fusion based approach resulted in a more robust classification. For instance, it could distinguish falls more accurately from certain fall-like activities, such as sitting down quickly and jumping, which is hard to detect using a single accelerometer. This work had inspired further research on sensor fusion. These two types of sensors can nowadays be found in all smart phones (Zhang et al., 2006; Dai et al., 2010; Abbate et al., 2012).

Besides the two non-vision based types of sensors mentioned above, vision-based sensors, such as surveillance cameras, and ambience-based, started becoming an attractive alternative. Rougier et al. (2011b) proposed a shape matching technique to track a person's silhouette through a video sequence. The deformation of the human shape is then quantified from the silhouettes based on shape analysis methods. Finally, falls are classified from normal activities using a Gaussian mixture model. After surveillance cameras, depth cameras also attracted substantial attention in this field. The earliest research which applied Time-of-Flight (TOF) depth camera was conducted in 2010 by Diraco et al. (2010). They proposed a novel approach based on visual sensors, which does not require landmarks, calibration patterns or user intervention. A ToF camera is, however, expensive and has low image resolution. Following that, the Kinect depth camera was first used in 2011 by Rougier et al. (2011a). Two features, human centroid height and velocity of body, were extracted from depth information. A simple threshold based algorithm was applied to detect falls and an overall success rate of 98.7% was achieved.

After the introduction of Kinect by Microsoft, there was a large shift in research from accelerometers to depth cameras. Accelerometers and depth cameras have become the most popular individual and combined sensors (Li et al., 2018). The combination of these two sensors achieved a substantial improvement when compared to the individual use of the sensors separately.

2.4. Strategy of the Literature Search

We use two databases, namely Web of Science and Google Scholar, to search for relevant literature. Since the sufficient advancements have been made at a rapid pace recently, searches included articles that were published in the last 6 years (since 2014). We also consider, all survey papers that were published on the topic of fall detection. Moreover, we give an account of all relevant benchmark data sets that have been used in this literature.

For the keywords "fall detection", 4,024 and 575,000 articles were found for the above two mentioned databases, respectively, since 2014. In order to narrow down our search to the more relevant articles we compiled a list of the most frequently used keywords that we report in **Table 1**.

We use the identified keywords above to generate the queries listed in **Table 2** in order to make the search more specific to the three classes of sensors that we are interested in. For the

TABLE 1 | The most frequently used keywords in the topic of fall detection.

Wearable sensor	Visual sensor	Ambient sensor	Sensor fusion
Fall detection	Fall detection	Fall detection	Fall detection
Falls	Falls	Falls	Falls
Fall accident	Fall accident	Fall accident	Fall accident
Machine learning	Machine learning	Machine learning	Machine learning
Deep learning	Deep learning	Deep learning	Deep learning
Reinforcement learning	Reinforcement learning	Reinforcement learning	Reinforcement learning
Body area networks	Multiple camera	Ambient sensor	Health monitoring
Wearable	Visual	Ambient	Sensor fusion
Worn	Vision-based	Ambience	Sensor network
Accelerometer	Kinect	RF-sensing	Data fusion
Gyroscope	Depth camera	WiFi	Multiple sensors
Biosensor	Video surveillance	Radar	Camera arrays
Smart watch	RGB camera	Cellular	Decision fusion
Gait	Infrared camera	Vibration	Anomaly detection
Wearable based	Health- monitoring	Ambience-based	IoT

They are manually classified into four categories.

TABLE 2 | Search queries used in Google Scholar and Web of Science for the three types of sensor and sensor fusion.

Sensor type	Query
Wearable-based	(Topic): ((“Fall detection" OR “Fall” OR “Fall accident”) AND (“Wearable” OR “Worn” OR “Accelerometer” OR “Machine learning” OR “Deep learning” OR “Reinforcement learning”) NOT “Survey” NOT “Review” NOT “Kinect” NOT “Video” NOT “Infrared” NOT “Ambient”)
Vision-based	(Topic): ((“Fall detection” OR “Falls” OR “Fall accident”) AND (“Video” OR “Visual” OR “Vision-based” OR “Kinect” OR “Depth camera” OR “Video surveillance” OR “RGB camera” OR “Infrared camera” OR “Monocular camera” OR “Machine learning” OR “Deep learning” OR “Reinforcement learning”) NOT “Wearable” NOT “Ambient”)
Ambient-based	(Topic): ((“Fall detection” OR “Falls” OR “Fall accident”) AND (“Ambient” OR “Ambient-based” OR “Ambience-based” OR “RF-sensing” OR “WiFi” OR “Cellular” OR “vibration” OR “Ambience” OR “Radar” OR “Machine learning” OR “Deep learning” OR “Reinforcement learning”) NOT “Wearable” NOT “vision”)
Sensor Fusion	(Topic): ((“Fall detection” OR “Falls” OR “Falls accident”) AND (“Health monitoring” OR “Multiple sensors” OR “Sensor fusion” OR “Sensor network” “Data fusion” OR “IoT” OR “Camera arrays” OR “Decision fusion” OR “Health monitoring” OR “Fusion” OR “Multiple sensors” OR “Machine learning” OR “Deep learning” OR “Reinforcement learning”))

retrieved articles, we discuss their contributions and keep only those that are truly relevant to our survey paper. For instance, articles that focus on rehabilitation after falls, and causes of falls, among others, are filtered out manually. This process, which is illustrated in **Figure 3**, ends up with a total of 87 articles, 13 of which describe benchmark data sets.

3. HARDWARE AND SOFTWARE COMPONENTS INVOLVED IN A FALL DETECTION SYSTEM

Most of the research of fall detection share a similar system architecture, which can be divided into four layers, namely Physiological Sensing Layer (PSL), Local Communication Layer (LCL), Information Processing Layer (IPL), and User application Layer (UAL), as suggested by Ray (2014) and illustrated in **Figure 4**.

PSL is the fundamental layer that contains various (smart) sensors used to collect physiological and ambient data from the persons being monitored. The most commonly used sensors nowadays include accelerometers that sense acceleration, gyroscopes that detect angular velocity, and magnetometers which sense orientation. Video surveillance cameras, which provide a more traditional means of sensing human activity, are also often used but are installed in specific locations, typically with fixed fields of views. More details about PSL are discussed in sections 4.1 and 5.1.

The next layer, namely LCL, is responsible for sending the sensor signals to the upper layers for further processing and analysis. This layer may have both wireless and wired methods of transmission, connected to local computing facilities or to cloud computing platforms. LCL typically takes the form of one (or potentially more) communication protocols, including wireless mediums like cellular, Zigbee, Bluetooth, WiFi, or even wired connections. We provide more details on LCL in sections 4.2 and 5.2.

IPL is a key component of the system. It includes hardware and software components, such as micro-controller, to analyze and transfer data from PSL to higher layers. In terms of software components, different kinds of algorithms, such as threshold, conventional machine learning, deep learning, and deep reinforcement learning are discussed in sections 4.3, 5.3, and 8.1.

Finally, the UAL concerns applications that assist the users. For instance, if a fall is detected in the IPL, a notification can first be sent to the user and if the user confirms the fall or does not answer, an alarm is sent to the nearest emergency caregivers who are expected to take immediate action. There are plenty of other products like Shimmer and AlertOne, which have been deployed as commercial applications to users. We also illustrate other different kinds of applications in section 7.

4. FALL DETECTION USING INDIVIDUAL SENSORS

4.1. Physiological Sensing Layer (PSL) of Individual Sensors

As mentioned above, fall detection research applied either a single sensor or fusion by multiple sensors. The methods of collecting data are typically divided into four main categories, namely

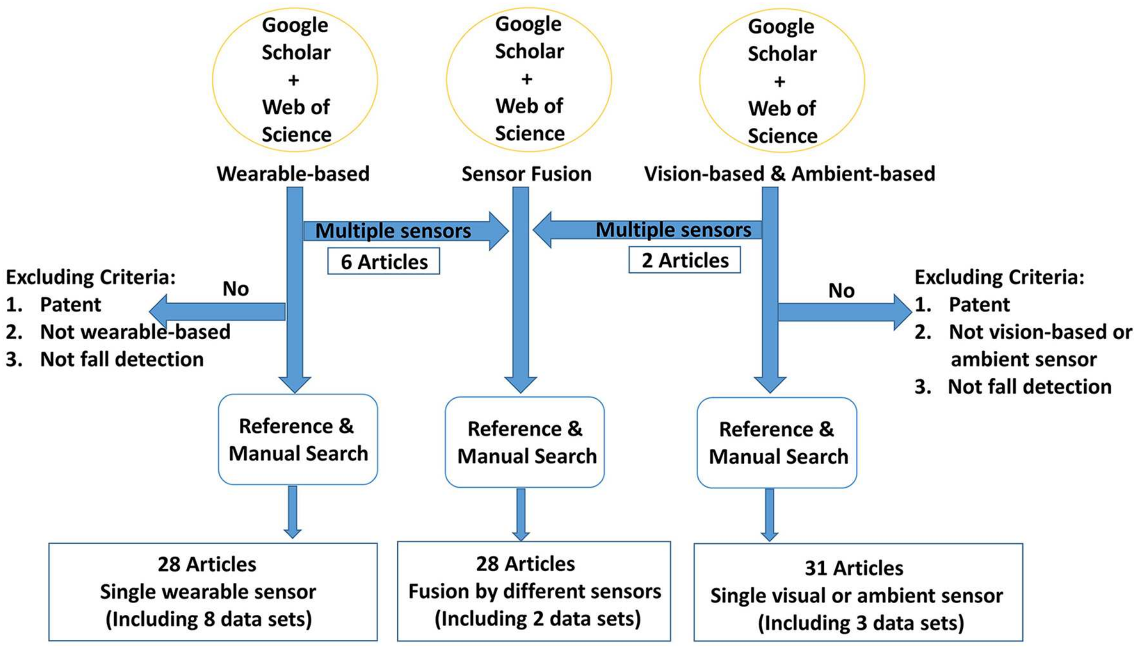

FIGURE 3 | Illustration of the literature search strategy. The wearable-based queries in **Table 2** return 28 articles. The vision- and ambient-based queries return 31 articles, and the sensor fusion queries return 28 articles.

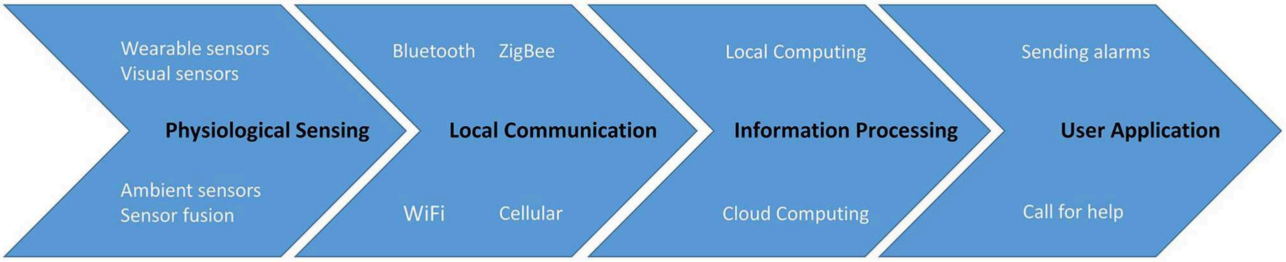

FIGURE 4 | The main components typically present within fall detection system architectures include the illustrated sequence of four layers. Data is collected in the physiological sensing layer, transferred through the local communication layer, then it is analyzed in the information processing layer, and finally the results are presented in the user application layer.

individual wearable sensors, individual visual sensors, individual ambient sensors and data fusion by sensor networks. Whilst some literature groups visual and ambient sensors together we treat them as two different categories in this survey paper due to visual sensors becoming more prominent as a detection method with the advent of depth cameras (RGBD), such as the Kinect.

4.1.1. Individual Wearable Sensors

Falls may result in key physiological variations of the human body, which provide a criterion to detect a fall. By measuring various human body related attributes using accelerometers, gyroscopes, glucometers, pressure sensors, ECG (Electrocardiography), EEG (Electroencephalography), or EOG (Electromyography), one can detect anomalies within subjects. Due to the advantages of mobility, portability, low cost, and availability, wearable devices are regarded as one of the key types of sensors for fall detection and have been widely studied. Numerous studies have been conducted to investigate wearable devices, which are regarded as a promising direction to study fall detection and prediction.

Based on our search criteria and filtering strategy (**Tables 1**, **2**), 28 studies, including eight papers focusing on public data sets, focusing on fall detection by individual wearable devices are selected and described to illustrate trends and challenges of fall detection during the past 6 years. Some conclusions can be drawn based on the literature during the past 6 years in comparison to the studies before 2014. From **Table 3**, we note that studies applying accelerometers account for a large percentage of research in this field. To the best of our knowledge, only Xi et al. (2017) deployed electromyography to detect falls, and 19 out of 20 papers applied an accelerometer to detect falls. Although the equipment used, such as Shimmer nodes,

TABLE 3 | Fall detection using individual wearable devices from 2014 to 2020.

References	Sensor	Location	No. subjects (age)	Data sets	Algorithms	Equipment	Alarm
Saleh and Jeannès (2019)	Accelerometer	Waist	23 (19–30), 15 (60–75)	Simulated	SVM	N/A	N
Zitouni et al. (2019)	Accelerometer	Sole	6 (N/A)	Simulated	Threshold	Smartsole	N/A
Thilo et al. (2019)	Accelerometer	Torso	15 (mean = 81)	N/A	N/A	N/A	Y
Wu et al. (2019)	Accelerometer	Chest and Thigh	42 (N/A), 36 (N/A)	Public (Simulated)	Decision tree	Smartwatch (Samsung watch)	N/A
Sucerquia et al. (2018)	Accelerometer	Waist	38 (N/A)	Public data sets			
Chen et al. (2018)	Accelerometer	Leg (pockets)	10 (20–26)	N/A	ML(SVM)	Smartphones	Y
Putra et al. (2017)	Accelerometer	Waist	38 (N/A), 42 (N/A)	Public data sets	ML	N	N/A
Khojasteh et al. (2018)	Accelerometer	N/A	17 (18–55), 6 (N/A), 15 (mean = 66.4)	Public (Simulated)	Threshold/ML	N/A	N/A
de Araújo et al. (2018)	Accelerometer	Wrist	1 (30)	N/A	Threshold	Smartwatch	N/A
Djelouat et al. (2017)	Accelerometer	Waist	N/A	Collected by authors (Simulated)	ML	Shimmer-3	Y
Aziz et al. (2017)	Accelerometer	Waist	10 (mean = 26.6)	Collected by authors (Simulated)	Threshold/ML	Accelerometers (Opal model, APDM Inc)	N
Kao et al. (2017)	Accelerometer	Wrist	N/A	Collected by authors (Simulated)	ML	ZenWatch(ASUS)	Y
Islam et al. (2017)	Accelerometer	Chest (pocket)	7 (N/A)	N/A	Threshold	Smartphone	N/A
Xi et al. (2017)	Electro-myography (sEMG)	Ankle, Leg	3 (24–26)	Collected by authors (Simulated)	ML	EMGworks 4.0 (DelSys Inc.)	N
Chen et al. (2017b)	Accelerometer	Lumbar, Thigh	22 (mean = 69.5)	Public data sets (Real)	ML	N/A	N/A
Chen et al. (2017b)	Accelerometer	Chest, Waist, Arm, Hand	N/A	Collected by authors (Simulated)	Threshold	N/A	Y
Medrano et al. (2017)	Accelerometer	N/A	10 (20–42)	Public (Simulated)	ML	Smartphones	N
Shi et al. (2016)	Accelerometer	N/A	10 (mean = 25)	N/A	Threshold	Smartphone	N/A
Wu et al. (2015)	Accelerometer	Waist	3 (23, 42, 60)	Collected by authors (Simulated)	Threshold	ADXL345 Accelerometer(ADI)	Y
Mahmud and Sirat (2015)	Accelerometer	Waist	13 (22–23)	Collected by authors (Simulated)	Threshold	Shimmer	N/A

ML is the abbreviation of Machine Learning.

TABLE 4 | Fall detection using individual vision-based devices from 2014 to 2020.

References	Sensor	No. subjects (age)	Data sets	Algorithms	Real-time	Alarm
Han et al. (2020)	Web camera	N/A	Simulated	CNN	N/A	N/A
Kong et al. (2019)	Camera (Surveillance)	N/A	Public (Simulated)	CNN	Y	N/A
Ko et al. (2018)	Camera (Smartphone)	N/A	Simulated	Rao-Blackwellized Particle Filtering	N/A	N
Shojaei-Hashemi et al. (2018)	Kinect	40 (10–15)	Public (Simulated)	LSTM	Y	N
Min et al. (2018)	Kinect	4 (N/A), 11 (22–39)	Public (Simulated)	SVM	Y	N
Ozcan et al. (2017)	Web camera	10 (24–31)	Simulated	Relative-entropy-based	N/A	N/A
Akagündüz et al. (2017)	Kinect	10 (N/A)	Public (Simulated) SDU (2011)	Silhouette	N/A	N
Adhikari et al. (2017)	Kinect	5 (19–50)	Simulated	CNN	N/A	N
Ozcan and Velipasalar (2016)	Camera (Smartphone)	10 (24–31)	Simulated	Threshold/ML	N/A	N/A
Senouci et al. (2016)	Web Camera	N/A	Simulated	SVM	Y	Y
Amini et al. (2016)	Kinect v2	11 (24–31)	Simulated	Adaptive Boosting Trigger, Heuristic	Y	N
Kumar et al. (2016)	Kinect	20 (N/A)	Simulated	SVM	N/A	N
Aslan et al. (2015)	Kinect	20 (N/A)	Public (Simulated)	SVM	N/A	N
Yun et al. (2015)	Kinect	12 (N/A)	Simulated	SVM	N/A	N
Stone and Skubic (2015)	Kinect	454 (N/A)	Public (Simulated+Real)	Decision trees	N/A	N
Bian et al. (2015)	Kinect	4 (24–31)	Simulated	SVM	N/A	N
Chua et al. (2015)	RGB camera	N/A	Simulated	Human shape variation	Y	N
Boulard et al. (2014)	Web camera	N/A	Real	Elliptical bounding box	N/A	N
Feng et al. (2014)	Monocular camera	N/A	Simulated	Multi-class SVM	Y	N
Mastorakis and Makris (2014)	Infrared sensor (Kinect)	N/A	Simulated	3D bounding box	Y	N
Gasparrini et al. (2014)	Kinect	N/A	Simulated	Depth frame analysis	Y	N
Yang and Lin (2014)	Kinect	N/A	Simulated	Silhouette	N/A	N

smartphones, and smart watches, often contain other sensors like gyroscopes and magnetometers, these sensors were not used to detect falls. Bourke et al. (2007) also found that accelerometers are regarded as the most popular sensors for fall detection mainly due to its affordable cost, easy installation and relatively good performance.

Although smartphones have gained attention for studying falls, the underlying sensors of systems using them are still accelerometers and gyroscopes (Shi et al., 2016; Islam et al., 2017; Medrano et al., 2017; Chen et al., 2018). Users are more likely to carry smartphones all day rather than extra wearable devices, so smartphones are useful for eventual real-world deployments (Zhang et al., 2006; Dai et al., 2010).

4.1.2. Individual Visual Sensors

Vision-based detection is another prominent method. Extensive effort in this direction has been demonstrated, and some of which (Akagündüz et al., 2017; Ko et al., 2018; Shojaei-Hashemi et al., 2018) show promising performance. Although most cameras are not as portable as wearable devices, they offer other advantages which deem them as decent options depending upon the scenario. Most static RGB cameras are not intrusive and wired hence there is no need to worry about battery limitations. Work on demonstrating viability of vision-based approaches have been demonstrated which makes use of infrared cameras (Mastorakis and Makris, 2014), RGB cameras (Charfi et al., 2012), and RGB-D depth cameras (Cai et al., 2017). One main challenge of vision-based detection is the potential violation of privacy due to the levels of detail that cameras can capture, such as personal information, appearance, and visuals of the living environment.

Further to the information that we report in **Table 4**, we note that RGB, depth, and infrared cameras are the three main visual sensors used. Moreover, it can be noted that the RGB-D camera (Kinect) is among the most popular vision-based sensor, as 12 out of 22 studies applied it in their work. Nine out of the other 10 studies used RGB cameras including cameras built into smartphones, web cameras, and monocular cameras, while the remaining study used an infrared camera within Kinect, to conduct their experiments.

Static RGB cameras are the most widely used sensors within the vision-based fall detection research conducted before 2004, although the accuracies of RGB camera-based detection systems vary drastically due to environmental conditions, such as illumination changes—which often results in limitations during the night. Besides, RGB cameras are inherently likely to have a higher false alarm rate because some deliberate actions like lying on the floor, sleeping or sitting down abruptly are not easily distinguished by frames captured by RGB cameras. With the launch of the Microsoft Kinect, which consists of an RGB camera, a depth sensor, and a multi-array microphone, it stimulated a trend in 3D data collection and analysis, causing a shift from RGB to RGB-D cameras. Kinect depth cameras took the place of the traditional RGB cameras and became the second popular sensors in the field of fall detection after 2014 (Xu et al., 2018).

In the last years, we are seeing an increased interest in the use of wearable cameras for the detection of falls. For instance, Ozcan and Velipasalar (2016) tried to exploit the

TABLE 5 | Fall detection using individual ambient devices from 2014 to 2020.

References	Sensor	No. subjects (age)	Data sets	Algorithms	Real-time	Alarm
Huang et al. (2019)	Vibration	12 (19-29)	Simulated	HMM	Y	N/A
Hao et al. (2019)	WiFi	N/A	Simulated	SVM	Y	N/A
Tian et al. (2018)	FMCW radio	140 (N/A)	Simulated	CNN	Y	N/A
Palipana et al. (2018)	WiFi	3 (27-30)	Simulated	SVM	Y	N/A
Wang et al. (2017a)	WiFi	6 (21-32)	Simulated	SVM	Y	N/A
Wang et al. (2017b)	WiFi	N/A	Simulated	SVM, Random Forests	N/A	N/A

cameras on smartphones. Smartphones were attached to the waists of subjects and their inbuilt cameras were used to record visual data. Ozcan et al. (2017) investigated how web cameras (e.g., Microsoft LifeCam) attached to the waists of subjects can contribute to fall detection. Although both approaches are not yet practical to be deployed in real applications, they show a new direction, which combines the advantages of wearable and visual sensors.

Table 4 reports the work conducted for individual vision-based sensors. The majority of research still makes use of simulated data. Only two studies use real world data; the one by Boulard et al. (2014) has actual fall data and the other by Stone and Skubic (2015) has mixed data, including 9 genuine falls and 445 simulated falls by trained stunt actors. In contrast to the real data sets from the work of Klenk et al. (2016) collected by wearable devices, there are few purely genuine data sets collected in real life scenarios using individual visual sensors.

4.1.3. Individual Ambient Sensors

The ambient sensor provides another non-intrusive means of fall detection. Sensors like active infrared, RFID, pressure, smart tiles, magnetic switches, Doppler Radar, ultrasonic, and microphone are used to detect the environmental changes due to falling as shown in **Table 5**. It provides an innovative direction in this field, which is passive and pervasive detection. Ultra-sonic sensor network systems are one of the earliest solutions in fall detection systems. Hori et al. (2004) argues that one can detect falls by putting a series of spatially distributed sensors in the space where elderly persons live. In Wang et al. (2017a,b), a new fall detection approach which uses ambient sensors is proposed. It relies on Wi-Fi which, due to its non-invasive and ubiquitous characteristics, is gaining more and more popularity. However, the studies by Wang et al. (2017a,b) are limited in terms of multi-person detection due to their classifiers not being robust enough to distinguish new subjects and environments. In order to tackle this issue, other studies have developed more sophisticated methods. These include the Aryokee (Tian et al., 2018) and FallDeFi (Palipana et al., 2018) systems. The Aryokee system is ubiquitous, passive and uses RF-sensing methods. Over 140 people were engaged to perform 40 kinds of activities in different environments for the collection of data and a convolutional neural network was utilized to classify falls. Palipana et al. (2018) developed a fall detection technique named FallDeFi, which is based on WiFi signals as the enabling sensing technology. They provided a system applying time-frequency of WiFi Channel State Information (CSI) and achieved above 93% average accuracy.

RF-sensing technologies have also been widely applied to other recognition activities beyond fall detection (Zhao et al., 2018; Zhang et al., 2019) and even for subtle movements. Zhao et al. (2018) studied human pose estimation with multiple persons. Their experiment showed that RF-pose has better performance under occlusion. This improvement is attributable to the ability of their method to estimate the pose of the subject through a wall, something that visual sensors fail to do. Further research on RF-sensing was conducted by Niu et al. (2018) with applications to finger gesture recognition, human respiration and chins movement. Their research can be potentially used for applications of autonomous health monitoring and home appliances control. Furthermore, Zhang et al. (2019) used an RF-sensing approach in the proposed system WiDIGR for gait recognition. Guo et al. (2019) claimed that RF-sensing is drawing more attention which can be attributed to being device-free for users, and in contrast to RGB cameras it can work under low light conditions and occlusions.

4.1.4. Subjects

For most research groups there is not enough time and funding to collect data continuously within several years to study fall detection. Due to the rarity of genuine data in fall detection and prediction, Li et al. (2013) have started to hire stunt actors to simulate different kinds of fall. There are also many data sets of falls which are simulated by young healthy students as indicated in the studies by Bourke et al. (2007) and Ma et al. (2014). For obvious reasons elderly subjects cannot be engaged to perform the motion of falls for data collection. For most of the existing data sets, falls are simulated by young volunteers who perform soft falls under the protection of soft mats on the ground. Elderly subjects, however, often have totally different behavior due to less control over the speed of the body. One potential solution could include simulated data sets created using physics engines, such as OpenSim. Previous research (Mastorakis et al., 2007, 2018) have shown that simulated data from OpenSim contributed to an increase in performance to the resulting models. Another solution includes online learning algorithms which adapt to subjects who were not represented in the training data. For instance, Deng et al. (2014) applied the Transfer learning reduced Kernel Extreme Learning Machine (RKELM) approach and showed how they can adapt a trained classifier—based on data sets collected by young volunteers—to the elderly.

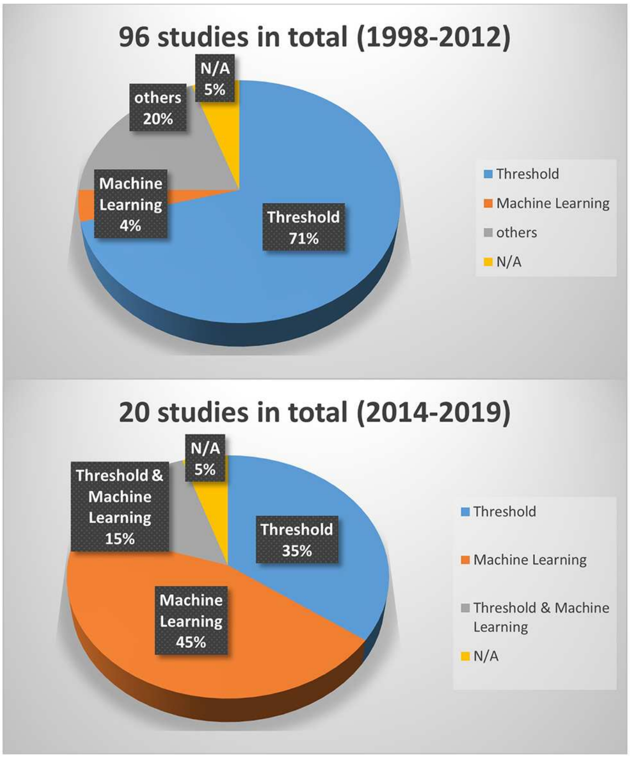

FIGURE 5 | Different types of methods used in fall detection using individual wearable sensors in the period 1998–2012 based on the survey of Schwickert et al. (2013) and in the period 2014–2020 based on our survey. The term "others" refers to traditional methods that are neither based on threshold nor on machine learning, and the term "N/A" stands for not available and refers to studies whose methods are not clearly defined.

The algorithm consists of two parts, namely offline classification modeling and online updating modeling, which is used to adapt to new subjects. After the model is trained by labeled training data offline, unlabeled test samples are fed into the pre-trained RKELM classifier and obtain a confidence score. The samples that obtain a confidence score above a certain threshold are used to update the model. In this way, the model is able to adapt to new subjects gradually when new samples are received from new subjects. Namba and Yamada (2018a,b) demonstrated how deep reinforcement learning can be applied to assisting mobile robots, in order to adapt to conditions that were not present in the training set.

4.2. Local Communication Layer (LCL) of Individual Sensors

There are two components which are involved with communication within such systems. Firstly, data collected from different smart sensors are sent to local computing facilities or remote cloud computing. Then, after the final decision is made by these computing platforms, instructions and alarms are sent to appointed caregivers for immediate assistance (El-Bendary et al., 2013).

Protocol of data communication is divided into two categories, namely wireless and wired transmission. For the former, transmission protocols include Zigbee, Bluetooth, Wifi, WiMax, and Cellular network.

Most of the studies that used individual wearable sensors deployed commercially available wearable devices. In those cases, data was communicated by transmission modules built in the wearable products, using mediums such as Bluetooth and cellular networks. In contrast to detection systems using wearable devices, most static vision- and ambient-based studies are connected to smart gateways by wired connections. These approaches are usually applied as static detection methods, so a wired connection is a better choice.

4.3. Information Processing Layer (IPL) of Individual Sensors

4.3.1. Detection Using Threshold-Based and Data-Driven Algorithms

Threshold-based and data-driven algorithms (including machine learning and deep learning) are the two main approaches that have been used for fall detection. Threshold-based approaches are usually used for data coming from individual sensors, such as accelerometers, gyroscopes, and electromyography. Their decisions are made by comparing measured values from concerned sensors to empirically established threshold values. Data driven approaches are more applicable for sensor fusion as they can learn non-trivial non-linear relationships from the data of all involved sensors. In terms of the algorithms used to analyze data collected using wearable devices, **Figure 5** demonstrates that there is a significant shift to machine learning based approaches, in comparison to the work conducted between 1998 and 2012. From papers presented between 1998 and 2012, threshold-based approaches account for 71%, while only 4% applied machine learning based methods (Schwickert et al., 2013). We believe that this shift is due to two main reasons. First, the rapid development of affordable sensors and the rise of the Internet-of-Things made it possible to more easily deploy multiple sensors in different applications. As mentioned above the non-linear fusion of multiple sensors can be modeled very well by machine learning approaches. Second, with the breakthrough of deep learning, threshold-based approaches have become even less preferable. Moreover, different types of machine learning approaches have been explored, namely, Bayesian networks, rule-based systems, nearest neighbor-based techniques, and neural networks. These data-driven approaches (Gharghan et al., 2018) show better accuracy and they are more robust in comparison to threshold-based methods. Notable is the fact that data-driven approaches are more resource hungry than threshold-based methods. With the ever advancement of technology, however, this is not a major concern and we foresee that more effort will be invested in this direction.

4.3.2. Detection Using Deep Learning

Traditional machine learning approaches determine mapping functions between extracted handcrafted features from raw training data and the respective output labels (e.g., no fall or fall, to keep it simple). The extraction of handcrafted features requires domain expertise and are, therefore, limited to the knowledge of the domain experts. Though such a limitation is imposed, literature shows that traditional machine learning, based on support vector machines, hidden Markov models, and decision trees are still very active in the field of fall detection that uses individual wearable non-visual or ambient sensors (e.g., accelerometer) (Wang et al., 2017a,b; Chen et al., 2018; Saleh and Jeannès, 2019; Wu et al., 2019). For visual sensors the trend has been moving toward deep learning for convolutional neural networks (CNN) (Adhikari et al., 2017; Kong et al., 2019; Han et al., 2020), or LSTM (Shojaei-Hashemi et al., 2018). Deep learning is a sophisticated learning framework that besides the mapping function (as mentioned above and used in traditional machine learning), it also learns the features (in a hierarchy fashion) that characterize the concerned classes (e.g., falls and no falls). This approach has been inspired by the visual system of the mammalian brain (LeCun et al., 2015). In computer vision applications, which take as input images or videos, deep learning has been established as state-of-the-art. In this regard, similar to other computer vision applications, fall detection approaches that rely on vision data have been shifting from traditional machine learning to deep learning in recent years.

4.3.3. Real Time and Alarms

Real-time is a key feature for fall detection systems, especially for commercial products. Considering that certain falls can be fatal or detrimental to the health, it is crucial that the deployed fall detection systems have high computational efficiency, preferably operating in (near) real-time. Below, we comment how the methods proposed in the reviewed literature fit within this aspect.

The percentage of studies applying real-time detection by static visual sensors are lower than that of wearable devices. For the studies using wearable devices, **Table 3** illustrates that six out of 20 studies that we reviewed can detect falls and send alarms. There are, however, few studies which demonstrate the ability to process data and send alerts in real-time for work conducted using individual visual sensors. Based on **Table 4**, one can note that although 40.9% (nine out of 22) of the studies claim that their systems can be used in real-time only one study showed that an alarm can actually be sent in real-time. The following are a couple of reasons why a higher percentage of vision-based systems can not be used in real time. Firstly, visual data is much larger and, therefore, its processing is more time consuming than that of one dimensional signals coming from non-vision-based wearable devices. Secondly, most of the work using vision sensors conducted their experiments with off-line methods, and modules like data transmission were not involved.

4.3.3.1. Summary

- For single-sensor-based fall detection systems most of the studies used data sets that include simulated falls by young and healthy volunteers. Further work is needed to establish whether such simulated falls can be used to detect genuine falls by the elderly.
- The types of sensors utilized in fall detection systems have changed in the past 6 years. For individual wearable sensors, accelerometers are still the most frequently deployed sensors. Static vision-based devices shifted from RGB to RGB-D cameras.
- Data-driven machine learning and deep learning approaches are gaining more popularity especially with vision-based systems. Such techniques may, however, be heavier than threshold-based counterparts in terms of computational resources.
- The majority of proposed approaches, especially those that rely on vision-based sensors, work in offline mode as they cannot operate in real-time. While such methods can be effective in terms of detection, their practical use is debatable as the time to respond is crucial.

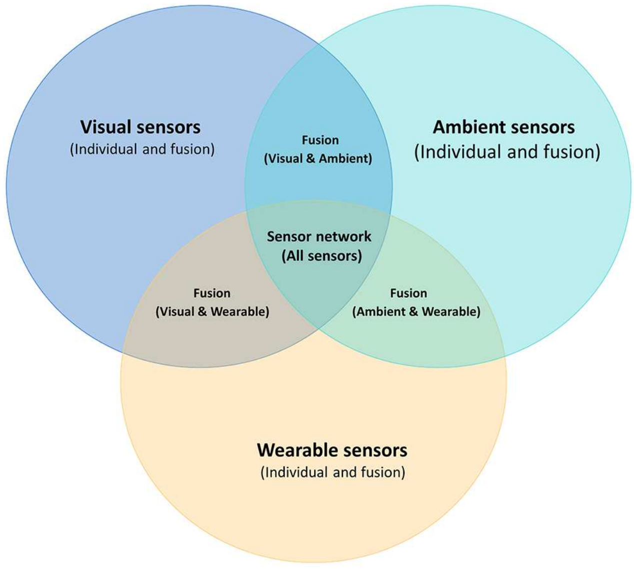

FIGURE 6 | Different kinds of individual sensors and sensor networks, including vision-based, wearable, and ambient sensors, along with sensor fusion.

5. SENSOR FUSION BY SENSOR NETWORK

5.1. Physiological Sensing Layer (PSL) Using Sensor Fusion

5.1.1. Sensors Deployed in Sensor Networks

In terms of sensor fusion, there are two categories, typically referred to as homogeneous and heterogeneous which take input from three types of sensors, namely wearable, visual, ambient sensors, as shown in **Figure 6**. Sensor fusion involves using multiple and different signals coming from various devices, which may for instance include, accelerometer, gyroscope, magnetometer, and visual sensors, among others. This is all done to complement the strengths of all devices for the design and development of more robust algorithms that can be used to monitor the health of subjects and detect falls (Spasova et al., 2016; Ma et al., 2019).

TABLE 6 | Fall detection by fusion of wearable sensors from 2014 to 2020.

Fusion within wearable sensors							
References	**Sensor**	**No. subjects (age)**	**Data sets**	**Algorithms**	**Real-time (Alarm)**	**Fusion method**	**Platforms**
Kerdjidj et al. (2020)	Accelerometer, Gyroscope	17 (N/A)	Simulated	Compressive sensing	Y (N/A)	Feature fusion	N/A
Xi et al. (2020)	Electromyography, Plantar Pressure	12 (23–27)	Simulated	FMMNN, DPK-OMELM	Y (Y)	Feature fusion	N/A
Chelli and Pätzold (2019)	Accelerometer, Gyroscope	30 (N/A)	Public (Simulated)	KNN, ANN, QSVM, EBT	Y (N/A)	Feature fusion	N/A
Queralta et al. (2019)	Accelerometer, Gyroscope, Magnetometer	57 (20-47)	Public (Simulated)	LTSM	Y(Y)	Feature fusion	N/A
Gia et al. (2018)	Accelerometer, Gyroscope, Magnetometer	2 (N/A)	N/A	Threshold	Y (Y)	Feature fusion	N/A
de Quadros et al. (2018)	Accelerometer, Gyroscope, Magnetometer	22 (mean = 26.09)	Simulated	Threshold/ML	N/A	Feature fusion	N/A
Yang et al. (2016)	Accelerometer, Gyroscope, Magnetometer	5 (N/A)	Simulated	SVM	Y (Y)	Feature fusion	PC
Pierleoni et al. (2015)	Accelerometer, Gyroscope, Magnetometer	10 (22–29)	Simulated	Threshold	Y (Y)	Feature fusion	ATmega328p (ATMEL)
Nukala et al. (2014)	Accelerometer, Gyroscopes	2 (N/A)	Simulated	ANN	Y (N/A)	Feature fusion	PC
Kumar et al. (2014)	Accelerometer, Pressure sensors, Heart rate monitor	N/A	Simulated	Threshold	Y (Y)	Partial fusion	PC
Hsieh et al. (2014)	Accelerometer, Gyroscope	3 (N/A)	Simulated	Threshold	N/A	Partial fusion	N/A

For the visual detection based approaches, the fusion of signals coming from RGB (Charfi et al., 2012), and RGB-D depth cameras along with camera arrays have been studied (Zhang et al., 2014). They showed that such fusion provides more viewpoints of detected locations, and improves the stability and robustness by decreasing false alarms due to occluded falls (Auvinet et al., 2011).

Li et al. (2018) combined accelerometer data from smartphones and Kinect depth data as well as smartphone camera signals. Liu et al. (2014) and Yazar et al. (2014) fused data from infrared sensors with ambient sensors, and data from doppler and vibration sensors separately. Among them, accelerometers and depth cameras (Kinect) are most frequently studied due to their low costs and effectiveness.

5.1.2. Sensor Networks Platform

Most of the existing IoT platforms, such as Microsoft Azure IoT, IBM Watson IoT Platform, and Google Cloud Platform, have not been used in the deployment of fall detection approaches by sensor fusion. In general, research studies on fall detection using sensor fusion are carried out by offline methods and decision fusion approaches. Therefore, in such studies, there is no need for data transmission and storage modules. From **Tables 6**, **7**, one can also observe that most of the time researchers applied their own workstations or personal computers as their platforms, as there was no need for the integration of sensors and real-time analysis in terms of fall detection in off-line mode.

Some works, such as those in Kwolek and Kepski (2014), Kepski and Kwolek (2014), and Kwolek and Kepski (2016), applied low-power single-board computer development platforms running in Linux, namely PandaBoard, PandaBoard ES, and A13-OlinuXino. A13-OlinuXino is an ARM-based single-board computer development platform, which runs Debian Linux distribution. PandaBoard ES, which is the updated version of PandaBoard, is a single-board computer development platform running at Linux. The PandaBoard ES can run different kinds of Linux-based operating systems, including Android and Ubuntu. It consists of 1 GB of DDR2 SDRAM, dual USB 2.0 ports as well as wired 10/100 Ethernet along with wireless Ethernet and Bluetooth connectivity. Linux is well-known for real-time embedded platforms since it provides various flexible inter-process communication methods, which is quite suitable for fall detection using sensor fusion.

In the research by Kwolek and Kepski (2014, 2016), wearable devices and Kinect were connected to the Pandaboard through Bluetooth and cable, separately. Firstly, data was collected by accelerometers and Kinect sensors, individually, which was then transmitted and stored in a memory card. The procedure of data transmission is asynchronous since there are different sampling rates for accelerometers and Kinect. Finally, all data was grouped together and processed by classification models that detected falls. The authors reported high accuracy rates but could not compare with other approaches since there is no benchmark data set.

Spasova et al. (2016) applied the A13-OlinuXino board as their platform. A standard web camera was connected to it via USB and an infrared camera was connected to the development board via I2C (Inter-Integrated Circuit). Their experiment achieved excellent performance with over 97% sensitivity and specificity. They claim that their system can be applied in real-time with hardware of low-cost and open source software platform.

TABLE 7 | Fall detection using fusion of sensor networks from 2014 to 2020.

References	Sensor	No. subjects (age)	Data sets	Algorithms	Real-time (Alarm)	Fusion method	Platforms
Fusion within visual sensors and ambient sensors							
Espinosa et al. (2019)	Two cameras	17 (18-24)	Simulated	CNN	N/A (N)	Feature fusion	N/A
Ma et al. (2019)	RGB camera, Thermal camera	14 (N/A)	Simulated	CNN	N/A (N)	Partial fusion	N/A
Spasova et al. (2016)	Web Camera, Infrared sensor	5 (27-81)	Simulated	SVM	Y (Y)	Partial fusion	A13-OlinuXino
Fusion within different kinds of individual sensors							
Martínez-Villaseñor et al. (2019)	Accelerometer, Gyroscope, Ambient light, Electroencephalograph, Infrared sensors, Web cameras	17 (18–24)	Simulated	Random Forest, SVM, ANN, kNN, CNN	Feature fusion	N/A	N/A
Li et al. (2018)	Accelerometer (smartphone), Kinect	N/A	Simulated	SVM, Threshold	Y (N/A)	Decision fusion	N/A
Daher et al. (2017)	Force sensors, Accelerometers	6 (N/A)	Simulated	Threshold	N (N/A)	Decision fusion	N/A
Ozcan and Velipasalar (2016)	Camera (smartphone), Accelerometer	10 (24 -30)	Simulated	Histogram of oriented gradients	Y (Y)	Decision fusion	N/A
Kwolek and Kepski (2016)	Accelerometer, Kinect	5 (N/A)	Simulated	Fuzzy logic	Y (Y)	Feature fusion, Partial fusion	PandaBoard ES
Sabatini et al. (2016)	Barometric altimeters, Accelerometer, Gyroscope	25 (mean = 28.3)	Simulated	Threshold	N/A (N)	Feature fusion	N/A
Chen et al. (2015)	Kinect, Inertial sensor	12 (23–30)	Public Simulated Ofli et al. (2013)	Collaborative representation,	N/A (N)	Feature fusion	N/A
Gasparrini et al. (2015)	Kinect v2, Accelerometer	11 (22-39)	Simulated	Threshold	N (N/A)	Data fusion	N/A
Kwolek and Kepski (2014)	Accelerometer, Kinect	5 (N/A)	Public (Simulated) URF (2014)	SVM, k-NN	Y (Y)	Partial fusion	PandaBoard ES
Kepski and Kwolek (2014)	Accelerometer, Kinect	30 (under 28)	Simulated	Alogorithms	Y (N)	Partial fusion	PandaBoard
Liu et al. (2014)	Passive infrared sensor, Doppler radar sensor	454 (N/A)	Simulated + Real life	SVM	N/A (N)	Decision fusion	N/A
Yazar et al. (2014)	Passive infrared sensors, Vibration sensor	N/A	Simulated	Threshold, SVM	N/A (N)	Decision fusion	N/A

Despite the available platforms mentioned above, the majority of fall detection studies trained their models in an offline mode with a single sensor on personal computers. The studies in Kwolek and Kepski (2014), Kepski and Kwolek (2014), Kwolek and Kepski (2016), and Spasova et al. (2016) utilized single-board computer platforms in their experiments to demonstrate the efficacy of their approaches. The crucial aspects of scalability and efficiency were not addressed and hence it is difficult to speculate the appropriateness of their methods in real-world applications. We believe that the future trend is to apply an interdisciplinary approach that deploys the data analysis modules on mature cloud platforms, which can provide a stable and robust environment while meeting the exploding demands of commercial applications.

5.1.3. Subjects and Data Sets

Although some groups devoted their efforts to acquire data of genuine falls, most researchers used data that contained simulated falls. We know that monitoring the lives of elderly people and waiting to capture real falls is very sensitive and time consuming. Having said that though, with regards to sensor fusion by wearable devices, there have been some attempts which have tried to build data sets of genuine data in real life. FARSEEING (Fall Repository for the design of Smart and self-adaptive Environments prolonging Independent living) is one such data set (Klenk et al., 2016). It is actually the largest data set of genuine falls in real life, and is open to public research upon request on their website. From 2012 to 2015, more than 2,000 volunteers have been involved, and more than

TABLE 8 | Comparison of different kinds of communication protocol.

Protocol	Zigbee	Bluetooth	WiFi	WiMax	Cellular network
Range	100 m	10 m	5 km	15 km	10–50 km
Data rate	250–500 kbps	1–3 Mbps	1–450 Mbps	75 Mbps	240 kbps
Band-width	2.4 GHz	2.4 GHz	2.4, 3.7, and 5 GHz	2.3, 2.5, and 3.5 GHz	824–894 MHz/1,900 MHz
Energy consumption	Low	Medium	High	N/A	N/A

300 real falls have been collected under the collaboration of six institutions[3].

As for the fusion by visual sensors and the combination of other non-wearable sensors, it becomes quite hard to acquire genuine data in real life. There was one group which tried to collect real data by visual sensors, but only nine real falls by elderly (Demiris et al., 2008) were captured during several years. The availability of only nine falls is too limited to train a meaningful model. As an alternative, Stone and Skubic (2015) hired trained stunt actors to simulate different kinds of falls and made a benchmark data set with 454 falls including 9 real falls by elderly.

5.2. Local Communication Layer (LCL) Using Sensor Fusion

Data transmission for fall detection using sensor networks can be done in different ways. In particular, Bluetooth (Pierleoni et al., 2015; Yang et al., 2016), Wi-Fi, ZigBee (Hsieh et al., 2014), cellular network using smart phones (Chen et al., 2018) and smart watches (Kao et al., 2017), as well as wired connection have all been explored. In studies that used wearable devices, most of them applied wireless methods, such as Bluetooth, which allowed the subject to move unrestricted.

Currently, when it comes to wireless sensors, Bluetooth has become probably the most popular communication protocol and it is widely used in existing commercial wearable products such as Shimmer. In the work by Yang et al. (2016), data is transmitted to a laptop in real-time by a Bluetooth module that is built in a commercial wearable device named Shimmer 2R. The sampling frame rate can be customized, and they chose to work with the 32-Hz sampling rate instead of the default sampling rate of 51.2-Hz. At high sampling frequencies, packet loss can occur and higher sampling rate also means higher energy consumption. Bluetooth is also applied to transmit data in non-commercial wearable devices. For example, Pierleoni et al. (2015) customized a wireless sensor node, where sensor module, micro-controller, Bluetooth module, battery, mass-storage unit, and wireless receiver were integrated within a prototype device of size 70–45–30 mm. Zigbee was used to transmit data in the work by Hsieh et al. (2014). In **Table 8**, we compare different kinds of wireless communication protocols.

[3] 1. Robert-Bosch Hospital (RBMF), Germany; 2. University of Tübingen, Germany; 3. University of Nürnberg/Erlangen, Germany; 4. German Sport University Cologne, Germany; 5. Bethanien-Hospital/Geriatric Center at the University of Heidelberg, Germany; 6. University of Auckland, New Zealand.

As for the data transmission using vision-based and ambient-based approaches, wired options are usually preferred. In the work by Spasova et al. (2016), a standard web camera was connected to an A13-OlinuXino board via USB and an infrared camera was connected to the development board via I2C (Inter-Integrated Circuit). Data and other messages were exchanged within the smart gateways through the internet.

For sensor fusion using different types of sensors, both wireless and cabled methods were utilized because of data variety. In the work by Kwolek and Kepski (2014, 2016), wearable devices and Kinect were connected to the Pandaboard through Bluetooth and cable, separately. Kinect was connected to a PC using USB interface and smart phones were connected by wireless methods (Li et al., 2018). These two types of sensor, smartphone and Kinect, were first used separately to monitor the same events and the underlying methods that processed their signals sent their output to a Netty server through the Internet where another method was used to fuse the outcomes of both methods to come to a final decision of whether the involved individual has fallen or not.

In the studies by Kwolek and Kepski (2014, 2016), accelerometers and Kinect cameras were connected to a pandaboard through Bluetooth and USB connections. Then, the final decision was made based on the data collected from the two sensors.

5.3. Information Processing Layer (IPL) Using Sensor Fusion

5.3.1. Methods of Sensor Fusion

Speaking of the fusion of different sensors, there are several criteria to group them. Yang and Yang (2006) and Tsinganos and Skodras (2018) grouped them into three categories, namely direct data fusion, feature fusion, and decision fusion. We divide sensor fusion techniques into four groups as shown in **Figure 7**, which we refer to as fusion with partial sensors, direct data fusion, feature fusion, and decision fusion.

For the partial fusion, although multiple sensors are deployed, only one sensor is used to take the final decision, such as the work by Ma et al. (2019). They used an RGB and a thermal camera to conduct their experiments, with the thermal camera being used only for the localization of faces. Falls were eventually detected only based on the data collected from the regular RGB cameras. A similar approach was applied by Spasova et al. (2016), where an infrared camera was deployed to confirm the presence of the subject and the data produced by the RGB camera was used to detect falls. There are also other works that used wearable devices that deployed the sensors at different stages. For instance,

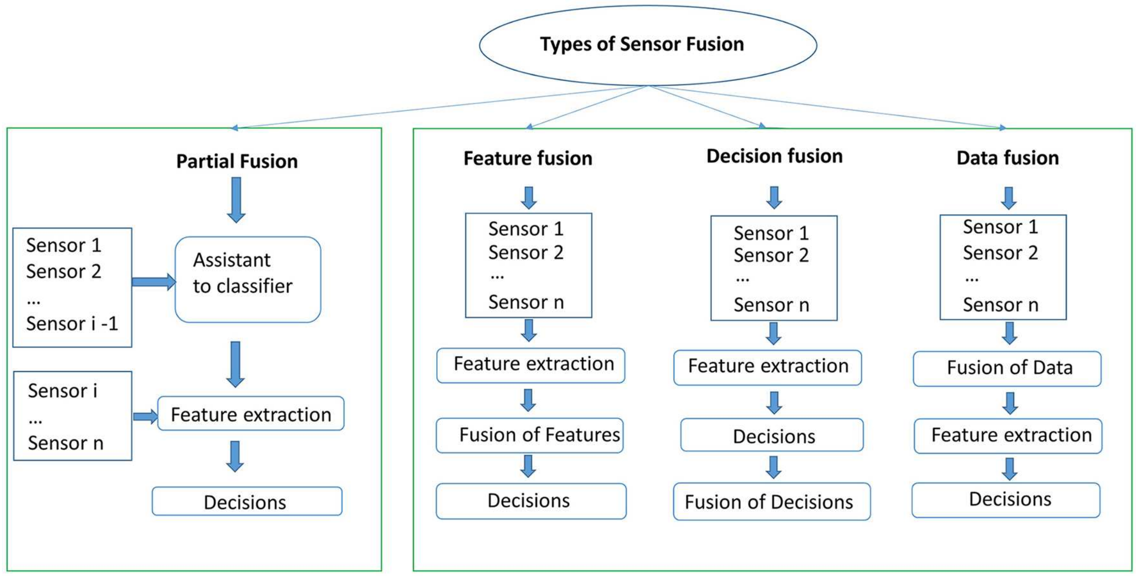

FIGURE 7 | Four kinds of sensor fusion methods including partial fusion, feature fusion, decision fusion, and data fusion. Partial fusion means that a subset of sensors are deployed to make decisions, while the other types of fusion techniques use all sensors as input.

in (Kepski and Kwolek, 2014; Kwolek and Kepski, 2014) a fall detection system was built by utilizing a tri-axial accelerometer and an RGB-D camera. The accelerometer was deployed to detect the motion of the subject. If the measured signal exceeded a given threshold then the Kinect was activated to capture the ongoing event.

The second approach of sensor fusion is known as feature fusion. In such an approach, feature extraction takes places on signals that come from different sensors. Then all features are merged into long feature vectors and used to train classification models. Most of the studies that we reviewed applied feature fusion for wearable-based fall detection systems. Many commercial products of wearable devices, sensors like accelerometers, gyroscope, magnetometer are built in one device. Data from these sensors is homogeneous synchronous with the same frequency and transmitted with built-in wireless modules. Having signals producing data with the synchronized frequency simplifies the fusion of data. Statistical features, such as mean, maximum, standard deviation, correlation, spectral entropy, spectral, sum vector magnitude, the angle between y-axis and vertical direction, and differential sum vector magnitude centroid can be determined from the signals of accelerometers, magnetometers, and gyroscopes, and used as features to train a classification model that can detect different types of falls (Yang et al., 2016; de Quadros et al., 2018; Gia et al., 2018).

Decision fusion is the third approach, where a chain of classifiers is used to come to a decision. A typical arrangement is to have a classification model that takes input from one type of sensor, another model that takes input from another sensor, and in turn the outputs of these two models are used as input to a third classification model that takes the final decision. Li et al. (2018) explored this approach with accelerometers embedded in smart phones and Kinect sensors. Ozcan and Velipasalar (2016) deployed an accelerometer and an RGB camera for the detection of falls. Different sensors, such as accelerometer, RGB and RGB-D cameras were deployed in these studies. Decisions are made separately based on the individual sensors, and then the final decision is achieved by combining the individual sensors.

The final approach is data fusion. This is achieved by first fusing the data from different sensors and perform feature extraction from the fused data. This is in contrast to feature fusion where data from these sensors is homogeneous synchronous with the same frequency. Data fusion can be applied to different sensors with different sampling frequency and data characteristics. Data from various sensors can be synchronized and combined directly for some sensors of different types. Because of the difference in sampling rate between the Kinect camera and wearable sensors, it is challenging to conduct feature fusion directly. In order to mitigate this difficulty, the transmission and exposure times of the Kinect camera are adapted to synchronize the RGB-D data with that of wearable sensors by an *ad-hoc* acquisition software, as was done by Gasparrini et al. (2015).

Ozcan and Velipasalar (2016) used both partial and feature fusion. They divided the procedure in two stages. In the first stage, only the accelerometer was utilized to indicate a potential fall, then the Kinect camera activates after the accelerometer flagged a potential fall. Features from both the Kinect camera and accelerometer were then extracted to classify activities of fall or non-fall in the second stage.

5.3.2. Machine Learning, Deep Learning, and Deep Reinforcement Learning

In terms of fall detection techniques based on wearable sensor fusion, the explored methods include threshold-based, traditional machine learning, and deep learning. The latter two are the most popular due to their robustness. The research by Chelli and Pätzold (2019) applied both traditional machine learning [kNN, QSVM, Ensemble Bagged Tree (EBT)] and deep learning. Their experiments were divided into two parts, namely activity recognition and fall detection. For the former, their experiments showed that traditional machine learning and deep learning outperformed other approaches, which showed 94.1 and 93.2% accuracy, respectively. Queralta et al. (2019) applied a long short-term memory (LSTM) approach, where wearable nodes including accelerometer, gyroscope, and magnetometer were embedded in a low power wide area network, with combined edge and fog computing. The LSTM algorithm is a type of recurrent neural network aimed at solving long sequence learning tasks. Their system achieved an average recall of 95% while providing a real-time solution of fall detection running on cloud platforms. Another example is the work by Nukala et al. (2014) who fused the measurements of accelerometers and gyroscopes and applied an Artificial Neural Network (ANN) for the modeling of fall detection.

As for visual sensor based fusion techniques, the limited studies that were included in our survey applied either traditional machine learning or deep learning (Espinosa et al., 2019; Ma et al., 2019) approaches. Fusion of multiple visual sensors from a public data set was presented by Espinosa et al. (2019), where a 2D CNN was trained to classify falls during daily life activities.

Another approach is reinforcement learning (RL), which is a growing branch in machine learning, and is gaining popularity in the fall detection field as well. Deep reinforcement learning (DRL) combines the advantages of deep learning and reinforcement learning, and has already shown its benefits in fall prevention (Namba and Yamada, 2018a,b; Yang, 2018) and fall detection (Yang, 2018). Namba and Yamada (2018a) proposed a fall risk prevention approach by assisting robots for the elderly living independently. Images and movies with the location information of accidents were collected. Most conventional machine learning and deep learning methods are, however, challenged when the operational environment changes. This is due to their data-driven nature that allows them to learn how to become robust mostly in the same environments where they were trained.

5.3.3. Data Storage and Analysis

Typical data storage devices include SD cards, local storage on the integration device, or remote storage on the cloud. For example, some studies used the camera and accelerometer in smartphones, and stored the data on the local storage of the smartphones (Ozcan and Velipasalar, 2016; Shi et al., 2016; Medrano et al., 2017). Other studies applied off-line methods and stored data in their own computer, and could be processed at a later stage. Alamri et al. (2013) argue that sensor-cloud will become the future trend because cloud platforms can be more open and more flexible than local platforms, which have limited local storage and processing power.

5.4. User Application Layer (UAL) of Sensor Fusion

Due to the rapid development of miniature bio-sensing devices, there has been a booming development of wearable sensors and other fall detection modules. Wearable modules, such as Shimmer, embedded with sensing sensors, communication protocols, and sufficient computational ability are available as affordable commercial products. For example, some wearable-based applications have been applied to the detection of falls and for monitoring health, in general. The target of the wearable devices is to wear and forget. Taking as an example the electronic skins (e-skins) that adhere to the body surface, clothing-based or accessory-based devices where proximity is sufficient. To fulfill the target of wearing and forgetting, many efforts have been put into the study of wearable systems, such as the My Heart project (Habetha, 2006), the Wearable Health Care System (WEALTHY) project (Paradiso et al., 2005), the Medical Remote Monitoring of clothes (MERMOTH) project (Luprano, 2006), and the project by Pandian et al. (2008). Some wearable sensors are also developed specifically to address fall detection. Shibuya et al. (2015) used a wearable wireless gait sensor for the detection of falls. More and more research work use existing commercial wearable products, which includes function of data transmission and sending alarms when falls are detected.

5.4.1. Summary

- Due to the sampling frequency and data characteristic, there are two main categories for sensor fusion. As shown in **Tables 6**, **7**, studies by sensor fusion are divided into fusion by sensor from the same category (e.g., fusion of wearable sensors, fusion of visual sensors, and fusion of ambient sensors) and fusion of sensors from different categories.
- Subjects in fall detection studies using sensor networks are still young and healthy volunteers, which is similar to that of individual sensors. Only one research adopted mixed data with simulated and genuine data.
- More wearable-based approaches are embedded with IoT platforms than that of vision-based approaches because data transmission and storage modules are built in existing commercial products.
- For the research combining sensors from different categories, the combination of accelerometer and Kinect camera is the most popular method.
- Partial fusion, data fusion, feature fusion, and decision fusion are four main methods of sensor fusion. Among them, feature fusion is the most popular approach, followed by decision fusion. For fusion using non-vision wearable sensors, most of the studies that we reviewed applied feature fusion, while decision fusion is the most appealing one for fusing sensors from different categories.

6. SECURITY AND PRIVACY

Because data generated by autonomous monitoring systems are security-critical and privacy-sensitive, there is an urgent demand to protect user's privacy and prevent these systems from being attacked. Cyberattacks on the autonomous monitoring systems may cause physical or mental damages and even threaten the lives of subjects under monitoring.

6.1. Security

In this survey we approached the systems of fall detection from different layers, including Physiological Sensing Layer (PSL), Local Communication Layer (LCL), Information Processing Layer (IPL), Internet Application Layer (IAL), and User Application Layer (UAL). Every layer faces security issues. For instance, information may leak in the LCL during data transmission, along with potential vulnerabilities with cloud storage and processing facility. Based on the literature that we report in **Tables 3–7**, most of the studies in the field of fall detection do not address security matters. Only few studies (Edgcomb and Vahid, 2012; Mastorakis and Makris, 2014; Ma et al., 2019) take privacy into consideration. Because of the distinct characteristics of wired and wireless transmission, it is still an open problem to find a comprehensive security protocol which can cover the security issues in both wired and wireless data transmission and storage (Islam et al., 2015).

6.2. Privacy

As mentioned above, privacy is one of the most important issue for users of autonomous health monitoring systems. Methods to protect privacy are dependent on the type of sensor used. Not all sensors tend to suffer from the issues of privacy equally. For example, vision-based sensors, like RGB cameras, are more vulnerable than wearable sensors, such as accelerometers, in terms of privacy. In the case of a detection system that uses only wearable sensors, problems of privacy are not as critical as systems involved with visual sensors.

In order to address the privacy concerns associated with RGB cameras some researchers proposed to mitigate them by blurring and distorting the appearances as post-processing steps in the application layer (Edgcomb and Vahid, 2012). An alternative way is to address the privacy issue in the design stage, as suggested by Ma et al. (2019). They investigated an optical level anonymous image sensing system. A thermal camera was deployed to locate faces and an RGB camera was used to detect falls. The location of the subject's face was used to generate a mask pattern on a spatial light modulator to control the light entering the RGB camera. Faces of subjects were blurred by blocking the visible light rays using the mask pattern on the spatial light modulator.

The infrared camera is another sensor which could protect the privacy of subjects. Mastorakis and Makris (2014) investigated an infrared camera built in a Kinect sensor. It only captures the thermal distribution of subjects and there is no information on the subject's appearance and living environment involved. Other vision-based sensors which could protect privacy are depth cameras. The fact they only capture depth information has made them more popular than RGB cameras.

As for the research of fall detection using sensor networks, different kinds of data are collected when more sensors are involved. Because of more data collection and transfer involved, the whole fall detection system by sensor fusion becomes more complicated and it makes the protection of privacy and security even harder. There is a trade-off between privacy and benefits of autonomous monitoring systems. The aim is to keep improving the algorithms while keeping the privacy and security issues to a minimum. This is the only way to make such systems socially acceptable.

7. PROJECTS AND APPLICATIONS AROUND FALL DETECTION

Approaches of fall detection evolve from personal emergency response systems (PERS) to intelligent automatic ones. One of the early fall detection systems sends an alarm by the PERS push-button, but it may fail when the concerned person loses consciousness or is too weak to move (Leff, 1997). Numerous attempts have been made to monitor not only falls but also other specific activities in autonomous health monitoring. Many projects have been conducted to develop applications of autonomous health monitoring, including fall detection, prediction, and prevention. Some of the aforementioned studies were promoted as commercial products. Different sensors from wearable sensors, visual sensors, and ambient sensors are deployed as commercial applications for fall detection. Among them, more wearable sensors have been developed as useful applications. For example, a company named Shimmer has developed 7 kinds of wearable sensing products aiming at autonomous health monitoring. One of the products is the Shimmer3 IMU Development Kit. It is a wearable sensor node including a sensing module, data transmission module, receiver, and it has been used by Mahmud and Sirat (2015) and Djelouat et al. (2017). The iLife fall detection sensor is developed by AlertOne[4], which provides the service of fall detection and one-button alert system. Smartwatch is another commercial solution for fall detection. Accelerometers embedded in smartwatches have been studied to detect falls (Kao et al., 2017; Wu et al., 2019). Moreover, Apple Watch Series 4 and later versions are equipped with the fall detection function, and it can help the consumer to connect to the emergency service. Although there are few specific commercial fall detection products based on RGB cameras, the relevant studies also show a promising future in the field. There are open source solutions provided by Microsoft using Kinect which could detect falls in real time and have the potential to be deployed as commercial products. As for ambient sensors, Linksys Aware apply tri-band mesh WiFi systems to fall detection, and they provide a premium subscription service as a commercial motion detection product. CodeBlue, a Harvard University research project, also focused on developing wireless sensor networks for medical applications (Lorincz et al., 2004). The MIThril project (DeVaul et al., 2003) is a next-generation wearable research platform developed by researchers at the MIT Media Lab. They made their software open source and hardware specifications available to the public.

The Ivy project (Pister et al., 2003) is a sensor network infrastructure from the Berkeley College of Engineering, University of California. The project aims to develop a sensor network system to provide assistance for the elderly living independently. Using a sensor network with fixed sensors and mobile sensors worn on the body, anomalies by the concerned elderly can be detected. Once falls are detected, the system sends alarms to caregivers to respond urgently.

A sensor network was built in 13 apartments in TigerPlace, which is an aging in place for people of retirement in Columbia, Missouri, and continuous data was collected for 3,339 days

[4]https://www.alert-1.com/

(Demiris et al., 2008). The sensor network with simple motion sensors, video sensors, and bed sensors that capture sleep restlessness and pulse and respiration levels, were installed in some apartments of 14 volunteers. Activities of 16 elderly people in TigerPlace, whose age range from 67 to 97, were recorded continuously and 9 genuine falls were captured. Based on the data set, Li et al. (2013) developed a sensor fusion algorithm. which achieved low rate of false alarms and a high detection rate.

8. TRENDS AND OPEN CHALLENGES

8.1. Trends

8.1.1. Sensor Fusion

There seems to be a general consensus that sensor fusion provides a more robust approach for the detection of elderly falls. The use of various sensors may complement each other in different situations. Thus, instead of relying on only one sensor, which may be unreliable if the conditions are not suitable for that sensor, the idea is to rely on different types of sensor that together can capture reliable data in various conditions. This results in a more robust system that can keep false alarms to a minimum while achieving high precision.

8.1.2. Machine Learning, Deep Learning and Deep Reinforcement Learning

Conventional machine learning approaches have been widely applied in fall detection and activity recognition, and results outperform those of threshold-based methods in studies that use wearable sensors. Deep learning is a subset of machine learning, which is concerned with artificial neural networks inspired by the mammalian brain. Approaches of deep learning are gaining popularity especially for visual sensors and sensor fusion and are becoming the state-of-the-art for fall detection and other activity recognition. Deep reinforcement learning is another promising research direction for fall detection. Reinforcement learning is inspired by the psychological neuro-scientific understandings of humans which can adapt and optimize decisions in a changing environment. Deep reinforcement learning combines advantages of deep learning, and reinforcement learning which can provide alternatives for detection that can adapt to the changing condition without sacrificing accuracy and robustness.

8.1.3. Fall Detection Systems on 5G Wireless Networks

5G is a softwarized and virtualized wireless network, which includes both a physical network and software virtual network functions. In comparison to 4G networks, 5th generation mobile introduces the ability of data transmission with high speed and low latency, which could contribute to the development of fall detection by IoT systems. Firstly, 5G is envisioned to become an important and universal communication protocol for IoT. Secondly, 5G cellular can be used for passive sensing approaches. Different from other kinds of RF-sensing approaches (e.g., WiFi or radar) which are aimed for short-distance indoor fall detection, the 5G wireless network can be applied to both indoor and outdoor scenarios as a pervasive sensing method. This type of network has already been successfully investigated by Gholampooryazdi et al. (2017) for the detection of crowd-size, presence detection, and walking speed, and their experiments showed accuracy of 80.9, 92.8, and 95%, respectively. Thirdly, we expect that 5G as a network is going to become a highly efficient and accurate platform to achieve better performance of anomaly detection. Smart networks or systems powered by 5G IoT and deep learning can be applied not only in fall detection systems, but also in other pervasive sensing and smart monitoring systems which assist elderly groups to live independently with high-quality life.

8.1.4. Personalized or Simulated Data

El-Bendary et al. (2013) and Namba and Yamada (2018b) have proposed to include historical medical and behavioral data of individuals along with sensor data. This allowed the enrichment of the data and consequently to make better informed decisions. This innovative perspective allows a more personalized approach as it uses the health profile of the concerned individual and it has the potential to become a trend also in this field. Another trend could be the way data sets are created to evaluate systems for fall detection. Mastorakis et al. (2007, 2018) applied the skeletal model simulated in Opensim, which is an open-source software developed by Stanford University. It can simulate different kinds of pre-defined skeletal models. They acquired 132 videos of different types of falls, and trained their own algorithms based on those models. The high results that they report indicate that the simulated falls by OpenSim are very realistic and, therefore, effective for training a fall detection model. Physics engines, like Opensim, can simulate customized data based on the height and age of different subjects and it offers the possibility of new directions to detect falls. Another solution, which can potentially address the scarcity of data, is to develop algorithms that can be adapted to subjects that were not part of the original training set (Deng et al., 2014; Namba and Yamada, 2018a,b) as we described in section 4.1.4.

8.1.5. Fog Computing

As to architecture is concerned, Fog computing offers the possibility to distribute different levels of processing across the involved edge devices in a decentralized way. Smart devices that can carry out some processing and that can communicate directly with each other are more attractive for (near) real-time processing as opposed to systems based on cloud computing (Queralta et al., 2019). An example of such smart devices include the Intel® RealSense™ depth camera, which includes a 28 nanometer (nm) processor to compute real-time depth images.

8.2. Open Challenges

The topic of fall detection has been studied extensively during the past two decades and many attempts have been proposed. The rapid development of new technologies keeps this topic very active in the research community. Although much progress has been made, there are still various open challenges, which we discuss below.

1. **The rarity of data of real falls**: There is no convincing public data set which could provide a gold standard. Many simulated data sets by individual sensors are available, but

it is debatable whether models trained on data collected by young and healthy subjects can be applied to elderly people in real-life scenarios. To the best of our knowledge, only Liu et al. (2014) used a data set with nine real falls along with 445 simulated ones. As for data sets with multiple sensors, the data sets are even scarcer. There is, therefore, an urgent need to create a benchmark data set of data coming from multiple sensors.

2. **Detection in real-time:** The attempts that we have seen in the literature are all based on offline methods that detect falls. While this is an important step, it is time that research starts focusing more on real-time systems that can be applied in the real-world.
3. **Security and privacy**: We have seen little attention to the security and privacy concerned with fall detection approaches. Security and privacy is therefore another topic which to our opinion must be addressed in cohesion with fall detection methods.
4. **Platform of sensor fusion:** It is still a novice topic with a lot of potential. Studies so far have treated this topic to a minimum as they mostly focused on the analytics aspect of the problem. In order to bring solutions closer to the market more holistic studies are needed to develop full information systems that can deal with the management and transmission of data in an efficient, effective and secure way.
5. **Limitation of location:** Some sensors, such as visual ones, have limited capability because they are fixed and static. It is necessary to develop fall detection systems which can be applied to controlled (indoor) and uncontrolled (outdoor) environments.
6. **Scalability and flexibility:** With the increasing number of affordable sensors there is a crucial necessity to study the scalability of fall detection systems especially when inhomogeneous sensors are considered (Islam et al., 2015). There is an increasing demand for scalable fall detection approaches that do not sacrifice robustness or security. When considering cloud-based trends, fall detection modules, such as data transmission, processing, applications, and services, should be configurable and scalable in order to adapt to the growth of commercial demands. Cloud-based systems enable more scalability of health monitoring systems at different levels as the need for resources of both hardware and software level changes with time. Cloud-based systems can add or remove sensors and services with little effort on the architecture (Alamri et al., 2013).

9. SUMMARY AND CONCLUSIONS

In this review we give an account on fall detection systems from a holistic point of view that includes data collection, data management, data transmission, security and privacy as well as applications.

In particular we compare approaches that rely on individual sensors with those that are based on sensor networks with various fusion techniques. The survey provides a description of the components of fall detection and it is aimed to give a comprehensive understanding of physical elements, software organization, working principles, techniques, and arrangement of different components that concern fall detection systems.

We draw the following conclusions.

1. The sensors and algorithms proposed during the past 6 years are very different in comparison to the research before 2014. Accelerometers are still the most popular sensors in wearable devices, while Kinect took the place of the RGB camera and became the most popular visual sensor. The combination of Kinect and accelerometer is turning out to be the most sought after.
2. There is not yet a benchmark data set on which fall detection systems can be evaluated and compared. This creates a hurdle in advancing the field. Although there has been an attempt to use middle-age subjects to simulate falls (Kangas et al., 2008), there are still differences in behavior between the elderly and middle-aged subjects.
3. Sensor fusion seems to be the way forward. It provides more robust solutions in fall detection systems but come with higher computational costs when compared to those that rely on individual sensors. The challenge is therefore to mitigate the computational costs.
4. Existing studies focus mainly on the data analytics aspect and do not give too much attention to IoT platforms in order to build full and stable systems. Moreover, the effort is put on analyzing data in offline mode. In order to bring such systems to the market, more effort needs to be invested in building all the components that make a robust, stable, and secure system that allows (near) real-time processing and that gains the trust of the elderly people.

The detection of elderly falls is an example of the potential of autonomous health monitoring systems. While the focus here was on elderly people, the same or similar systems can be applicable to people with mobility problems. With the ongoing development of IoT devices, autonomous health monitoring and assistance systems that rely on such devices seems to be the key for the detection of early signs of physical and cognitive problems that can range from cardiovascular issues to mental disorders, such as Alzheimer's and dementia.

AUTHOR CONTRIBUTIONS

GA and XW conceived and planned the paper. XW wrote the manuscript in consultation with GA and JE. All authors listed in this paper have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

XW holds a fellowship (grant number: 201706340160) from the China Scholarship Council supplemented by the University of Groningen. The support provided by the China Scholarship Council (CSC) during the study at the University of Groningen is acknowledged.

REFERENCES

(2011). *Sdufall*. Available online at: http://www.sucro.org/homepage/wanghaibo/SDUFall.html

(2014). *Urfd*. Available online at: https://sites.google.com/view/haibowang/home

Abbate, S., Avvenuti, M., Bonatesta, F., Cola, G., Corsini, P., and Vecchio, A. (2012). A smartphone-based fall detection system. *Pervas. Mobile Comput*. 8, 883–899. doi: 10.1016/j.pmcj.2012.08.003

Adhikari, K., Bouchachia, H., and Nait-Charif, H. (2017). "Activity recognition for indoor fall detection using convolutional neural network," in *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)* (Nagoya: IEEE), 81–84. doi: 10.23919/MVA.2017.7986795

Akagündüz, E., Aslan, M., Şengür, A., Wang, H., and İnce, M. C. (2017). Silhouette orientation volumes for efficient fall detection in depth videos. *IEEE J. Biomed. Health Inform*. 21, 756–763. doi: 10.1109/JBHI.2016.2570300

Alamri, A., Ansari, W. S., Hassan, M. M., Hossain, M. S., Alelaiwi, A., and Hossain, M. A. (2013). A survey on sensor-cloud: architecture, applications, and approaches. *Int. J. Distribut. Sensor Netw*. 9, 917923. doi: 10.1155/2013/917923

Amini, A., Banitsas, K., and Cosmas, J. (2016). "A comparison between heuristic and machine learning techniques in fall detection using kinect v2," in *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)* (Benevento: IEEE), 1–6. doi: 10.1109/MeMeA.2016.7533763

Aslan, M., Sengur, A., Xiao, Y., Wang, H., Ince, M. C., and Ma, X. (2015). Shape feature encoding via fisher vector for efficient fall detection in depth-videos. *Applied Soft. Comput.* 37, 1023–1028. doi: 10.1016/j.asoc.2014.12.035

Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., and Meunier, J. (2011). Fall detection with multiple cameras: an occlusion-resistant method based on 3-D silhouette vertical distribution. *IEEE Trans. Inform. Technol. Biomed*. 15, 290–300. doi: 10.1109/TITB.2010.2087385

Aziz, O., Musngi, M., Park, E. J., Mori, G., and Robinovitch, S. N. (2017). A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials. *Med. Biol. Eng. Comput*. 55, 45–55. doi: 10.1007/s11517-016-1504-y

Bian, Z.-P., Hou, J., Chau, L.-P., and Magnenat-Thalmann, N. (2015). Fall detection based on body part tracking using a depth camera. *IEEE J. Biomed. Health Inform*. 19, 430–439. doi: 10.1109/JBHI.2014.2319372

Bloom, D. E., Boersch-Supan, A., McGee, P., and Seike, A. (2011). Population aging: facts, challenges, and responses. *Benefits Compens. Int*. 41, 22.

Boulard, L., Baccaglini, E., and Scopigno, R. (2014). "Insights into the role of feedbacks in the tracking loop of a modular fall-detection algorithm," in *2014 IEEE Visual Communications and Image Processing Conference* (Valletta: IEEE), 406–409. doi: 10.1109/VCIP.2014.7051592

Bourke, A., O'brien, J., and Lyons, G. (2007). Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait Post*. 26, 194–199. doi: 10.1016/j.gaitpost.2006.09.012

Bourke, A. K., and Lyons, G. M. (2008). A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Med. Eng. Phys*. 30, 84–90. doi: 10.1016/j.medengphy.2006.12.001

Cai, Z., Han, J., Liu, L., and Shao, L. (2017). RGB-D datasets using Microsoft Kinect or similar sensors: a survey. *Multimedia Tools Appl*. 76, 4313–4355. doi: 10.1007/s11042-016-3374-6

Charfi, I., Miteran, J., Dubois, J., Atri, M., and Tourki, R. (2012). Definition and performance evaluation of a robust SVM based fall detection solution. *SITIS* 12, 218–224. doi: 10.1109/SITIS.2012.155

Chaudhuri, S., Thompson, H., and Demiris, G. (2014). Fall detection devices and their use with older adults: a systematic review. *J. Geriatr. Phys. Ther*. 37, 178. doi: 10.1519/JPT.0b013e3182abe779

Chelli, A., and Pätzold, M. (2019). A machine learning approach for fall detection and daily living activity recognition. *IEEE Access* 7, 38670–38687. doi: 10.1109/ACCESS.2019.2906693

Chen, C., Jafari, R., and Kehtarnavaz, N. (2015). "UTD-MHAD: a multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *2015 IEEE International Conference on Image Processing (ICIP)* (Quebec City: IEEE), 168–172. doi: 10.1109/ICIP.2015.7350781

Chen, C., Jafari, R., and Kehtarnavaz, N. (2017a). A survey of depth and inertial sensor fusion for human action recognition. *Multimedia Tools Appl*. 76, 4405–4425. doi: 10.1007/s11042-015-3177-1

Chen, K.-H., Hsu, Y.-W., Yang, J.-J., and Jaw, F.-S. (2017b). Enhanced characterization of an accelerometer-based fall detection algorithm using a repository. *Instrument. Sci. Technol*. 45, 382–391. doi: 10.1080/10739149.2016.1268155

Chen, K.-H., Hsu, Y.-W., Yang, J.-J., and Jaw, F.-S. (2018). Evaluating the specifications of built-in accelerometers in smartphones on fall detection performance. *Instrument. Sci. Technol*. 46, 194–206. doi: 10.1080/10739149.2017.1363054

Chua, J.-L., Chang, Y. C., and Lim, W. K. (2015). A simple vision-based fall detection technique for indoor video surveillance. *Signal Image Video Process*. 9, 623–633. doi: 10.1007/s11760-013-0493-7

Daher, M., Diab, A., El Najjar, M. E. B., Khalil, M. A., and Charpillet, F. (2017). Elder tracking and fall detection system using smart tiles. *IEEE Sens. J*. 17, 469–479. doi: 10.1109/JSEN.2016.2625099

Dai, J., Bai, X., Yang, Z., Shen, Z., and Xuan, D. (2010). "PerfallD: a pervasive fall detection system using mobile phones," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (Mannheim: IEEE), 292–297.

de Araújo, Í. L., Dourado, L., Fernandes, L., Andrade, R. M. C., and Aguilar, P. A. C. (2018). "An algorithm for fall detection using data from smartwatch," in *2018 13th Annual Conference on System of Systems Engineering (SoSE)* (Paris: IEEE), 124–131. doi: 10.1109/SYSOSE.2018.8428786

de Quadros, T., Lazzaretti, A. E., and Schneider, F. K. (2018). A movement decomposition and machine learning-based fall detection system using wrist wearable device. *IEEE Sens. J*. 18, 5082–5089. doi: 10.1109/JSEN.2018.2829815

Demiris, G., Hensel, B. K., Skubic, M., and Rantz, M. (2008). Senior residents' perceived need of and preferences for "smart home" sensor technologies. *Int. J. Technol. Assess. Health Care* 24, 120–124. doi: 10.1017/S0266462307080154

Deng, W.-Y., Zheng, Q.-H., and Wang, Z.-M. (2014). Cross-person activity recognition using reduced kernel extreme learning machine. *Neural Netw*. 53, 1–7. doi: 10.1016/j.neunet.2014.01.008

DeVaul, R., Sung, M., Gips, J., and Pentland, A. (2003). "Mithril 2003: applications and architecture," in *Null* (White Plains, NY: IEEE), 4. doi: 10.1109/ISWC.2003.1241386

Diraco, G., Leone, A., and Siciliano, P. (2010). "An active vision system for fall detection and posture recognition in elderly healthcare," in *2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)* (Dresden: IEEE), 1536–1541. doi: 10.1109/DATE.2010.5457055

Djelouat, H., Baali, H., Amira, A., and Bensaali, F. (2017). "CS-based fall detection for connected health applications," in *2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME)* (Beirut: IEEE), 1–4. doi: 10.1109/ICABME.2017.8167540

Edgcomb, A., and Vahid, F. (2012). Privacy perception and fall detection accuracy for in-home video assistive monitoring with privacy enhancements. *ACM SIGHIT Rec*. 2, 6–15. doi: 10.1145/2384556.2384557

El-Bendary, N., Tan, Q., Pivot, F. C., and Lam, A. (2013). Fall detection and prevention for the elderly: a review of trends and challenges. *Int. J. Smart Sens. Intell. Syst*. 6. doi: 10.21307/ijssis-2017-588

Espinosa, R., Ponce, H., Gutiérrez, S., Martínez-Villaseñor, L., Brieva, J., and Moya-Albor, E. (2019). A vision-based approach for fall detection using multiple cameras and convolutional neural networks: a case study using the up-fall detection dataset. *Comput. Biol. Med*. 115:103520. doi: 10.1016/j.compbiomed.2019.103520

Feng, W., Liu, R., and Zhu, M. (2014). Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera. *Signal Image Video Process*. 8, 1129–1138. doi: 10.1007/s11760-014-0645-4

Gasparrini, S., Cippitelli, E., Gambi, E., Spinsante, S., Wåhslén, J., Orhan, I., et al. (2015). "Proposal and experimental evaluation of fall detection solution based on wearable and depth data fusion," in *International Conference on ICT Innovations* (Ohrid: Springer), 99–108. doi: 10.1007/978-3-319-25733-4_11

Gasparrini, S., Cippitelli, E., Spinsante, S., and Gambi, E. (2014). A depth-based fall detection system using a kinect® sensor. *Sensors* 14, 2756–2775. doi: 10.3390/s140202756

Gharghan, S., Mohammed, S., Al-Naji, A., Abu-AlShaeer, M., Jawad, H., Jawad, A., et al. (2018). Accurate fall detection and localization for elderly people based on neural network and energy-efficient wireless sensor network. *Energies* 11, 2866. doi: 10.3390/en11112866

Gholampooryazdi, B., Singh, I., and Sigg, S. (2017). "5G ubiquitous sensing: passive environmental perception in cellular systems," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)* (Toronto: IEEE), 1–6. doi: 10.1109/VTCFall.2017.8288261

Gia, T. N., Sarker, V. K., Tcarenko, I., Rahmani, A. M., Westerlund, T., Liljeberg, P., et al. (2018). Energy efficient wearable sensor node for iot-based fall detection systems. *Microprocess. Microsyst*. 56, 34–46. doi: 10.1016/j.micpro.2017.10.014

Guo, B., Zhang, Y., Zhang, D., and Wang, Z. (2019). Special issue on device-free sensing for human behavior recognition. *Pers. Ubiquit. Comput*. 23, 1–2. doi: 10.1007/s00779-019-01201-8

Habetha, J. (2006). "The myheart project-fighting cardiovascular diseases by prevention and early diagnosis," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE* (New York, NY: IEEE), 6746–6749. doi: 10.1109/IEMBS.2006.260937

Han, Q., Zhao, H., Min, W., Cui, H., Zhou, X., Zuo, K., et al. (2020). A two-stream approach to fall detection with mobileVGG. *IEEE Access* 8, 17556–17566. doi: 10.1109/ACCESS.2019.2962778

Hao, Z., Duan, Y., Dang, X., and Xu, H. (2019). "KS-fall: Indoor human fall detection method under 5GHZ wireless signals," in *IOP Conference Series: Materials Science and Engineering*, Vol. 569 (Sanya: IOP Publishing), 032068. doi: 10.1088/1757-899X/569/3/032068

Hori, T., Nishida, Y., Aizawa, H., Murakami, S., and Mizoguchi, H. (2004). "Sensor network for supporting elderly care home," in *Sensors, 2004, Proceedings of IEEE* (Vienna: IEEE), 575–578. doi: 10.1109/ICSENS.2004.1426230

Hsieh, S.-L., Chen, C.-C., Wu, S.-H., and Yue, T.-W. (2014). "A wrist-worn fall detection system using accelerometers and gyroscopes," in *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control* (Miami: IEEE), 518–523. doi: 10.1109/ICNSC.2014.6819680

Huang, Y., Chen, W., Chen, H., Wang, L., and Wu, K. (2019). "G-fall: device-free and training-free fall detection with geophones," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)* (Boston, MA: IEEE), 1–9. doi: 10.1109/SAHCN.2019.8824827

Igual, R., Medrano, C., and Plaza, I. (2013). Challenges, issues and trends in fall detection systems. *Biomed. Eng. Online* 12, 66. doi: 10.1186/1475-925X-12-66

Islam, S. R., Kwak, D., Kabir, M. H., Hossain, M., and Kwak, K.-S. (2015). The internet of things for health care: a comprehensive survey. *IEEE Access* 3, 678–708. doi: 10.1109/ACCESS.2015.2437951

Islam, Z. Z., Tazwar, S. M., Islam, M. Z., Serikawa, S., and Ahad, M. A. R. (2017). "Automatic fall detection system of unsupervised elderly people using smartphone," in *5th IIAE International Conference on Intelligent Systems and Image Processing 2017* (Hawaii), 5. doi: 10.12792/icisip2017.077

Kangas, M., Konttila, A., Lindgren, P., Winblad, I., and Jämsä, T. (2008). Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait Post*. 28, 285–291. doi: 10.1016/j.gaitpost.2008.01.003

Kao, H.-C., Hung, J.-C., and Huang, C.-P. (2017). "GA-SVM applied to the fall detection system," in *2017 International Conference on Applied System Innovation (ICASI)* (Sapporo: IEEE), 436–439. doi: 10.1109/ICASI.2017.7988446

Kepski, M., and Kwolek, B. (2014). "Fall detection using ceiling-mounted 3D depth camera," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, Vol. 2 (Lisbon: IEEE), 640–647.

Kerdjidj, O., Ramzan, N., Ghanem, K., Amira, A., and Chouireb, F. (2020). Fall detection and human activity classification using wearable sensors and compressed sensing. *J. Ambient Intell. Human. Comput*. 11, 349–361. doi: 10.1007/s12652-019-01214-4

Khojasteh, S., Villar, J., Chira, C., González, V., and de la Cal, E. (2018). Improving fall detection using an on-wrist wearable accelerometer. *Sensors* 18:1350. doi: 10.3390/s18051350

Klenk, J., Schwickert, L., Palmerini, L., Mellone, S., Bourke, A., Ihlen, E. A., et al. (2016). The farseeing real-world fall repository: a large-scale collaborative database to collect and share sensor signals from real-world falls. *Eur. Rev. Aging Phys. Activity* 13:8. doi: 10.1186/s11556-016-0168-9

Ko, M., Kim, S., Kim, M., and Kim, K. (2018). A novel approach for outdoor fall detection using multidimensional features from a single camera. *Appl. Sci.* 8:984. doi: 10.3390/app8060984

Kong, Y., Huang, J., Huang, S., Wei, Z., and Wang, S. (2019). Learning spatiotemporal representations for human fall detection in surveillance video. *J. Visual Commun. Image Represent*. 59, 215–230. doi: 10.1016/j.jvcir.2019.01.024

Kumar, D. P., Yun, Y., and Gu, I. Y.-H. (2016). "Fall detection in RGB-D videos by combining shape and motion features," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Shanghai: IEEE), 1337–1341. doi: 10.1109/ICASSP.2016.7471894

Kumar, S. V., Manikandan, K., and Kumar, N. (2014). "Novel fall detection algorithm for the elderly people," in *2014 International Conference on Science Engineering and Management Research (ICSEMR)* (Shanghai: IEEE), 1–3. doi: 10.1109/ICSEMR.2014.7043578

Kwolek, B., and Kepski, M. (2014). Human fall detection on embedded platform using depth maps and wireless accelerometer. *Comput. Methods Programs Biomed*. 117, 489–501. doi: 10.1016/j.cmpb.2014.09.005

Kwolek, B., and Kepski, M. (2016). Fuzzy inference-based fall detection using kinect and body-worn accelerometer. *Appl. Soft Comput*. 40, 305–318. doi: 10.1016/j.asoc.2015.11.031

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Leff, B. (1997). Persons found in their homes helpless or dead. *J. Am. Geriatr. Soc.* 45, 393–394. doi: 10.1111/j.1532-5415.1997.tb03788.x

Li, Q., Stankovic, J. A., Hanson, M. A., Barth, A. T., Lach, J., and Zhou, G. (2009). "Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information," in *2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks* (Berkeley, CA: IEEE), 138–143. doi: 10.1109/BSN.2009.46

Li, X., Nie, L., Xu, H., and Wang, X. (2018). "Collaborative fall detection using smart phone and kinect," in *Mobile Networks and Applications*, eds H. Janicke, D. Katsaros, T. J. Cruz, Z. M. Fadlullah, A.-S. K. Pathan, K. Singh et al. (Springer), 1–14. doi: 10.1007/s11036-018-0998-y

Li, Y., Banerjee, T., Popescu, M., and Skubic, M. (2013). "Improvement of acoustic fall detection using kinect depth sensing," in *2013 35th Annual International Conference of the IEEE Engineering in medicine and biology society (EMBC)* (Osaka: IEEE), 6736–6739.

Liu, L., Popescu, M., Skubic, M., and Rantz, M. (2014). "An automatic fall detection framework using data fusion of Doppler radar and motion sensor network," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Chicago, IL: IEEE), 5940–5943.

Lord, C. J., and Colvin, D. P. (1991). "Falls in the elderly: detection and assessment," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Orlando, FL: IEEE), 1938–1939.

Lorincz, K., Malan, D. J., Fulford-Jones, T. R., Nawoj, A., Clavel, A., Shnayder, V., et al. (2004). Sensor networks for emergency response: challenges and opportunities. *IEEE Pervas. Comput*. 3, 16–23. doi: 10.1109/MPRV.2004.18

Luprano, J. (2006). "European projects on smart fabrics, interactive textiles: Sharing opportunities and challenges," in *Workshop Wearable Technol. Intel. Textiles* (Helsinki).

Ma, C., Shimada, A., Uchiyama, H., Nagahara, H., and Taniguchi, R.-i. (2019). Fall detection using optical level anonymous image sensing system. *Optics Laser Technol*. 110, 44–61. doi: 10.1016/j.optlastec.2018.07.013

Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., and Li, Y. (2014). Depth-based human fall detection via shape features and improved extreme learning machine. *IEEE J. Biomed. Health Inform*. 18, 1915–1922. doi: 10.1109/JBHI.2014.2304357

Mahmud, F., and Sirat, N. S. (2015). Evaluation of three-axial wireless-based accelerometer for fall detection analysis. *Int. J. Integr. Eng*. 7, 15–20.

Martínez-Villaseñor, L., Ponce, H., Brieva, J., Moya-Albor, E., Núñez-Martínez, J., and Peñafort-Asturiano, C. (2019). Up-fall detection dataset: a multimodal approach. *Sensors* 19:1988. doi: 10.3390/s19091988

Mastorakis, G., Ellis, T., and Makris, D. (2018). Fall detection without people: a simulation approach tackling video data scarcity. *Expert Syst. Appl*. 112, 125–137. doi: 10.1016/j.eswa.2018.06.019

Mastorakis, G., Hildenbrand, X., Grand, K., and Makris, D. (2007). Customisable fall detection: a hybrid approach using physics based simulation and machine learning. *IEEE Trans. Biomed. Eng*. 54, 1940–1950.

Mastorakis, G., and Makris, D. (2014). Fall detection system using kinect's infrared sensor. *J. Realtime Image Process*. 9, 635–646. doi: 10.1007/s11554-012-0246-9

Medrano, C., Igual, R., García-Magariño, I., Plaza, I., and Azuara, G. (2017). Combining novelty detectors to improve accelerometer-based fall detection. *Med. Biol. Eng. Comput*. 55, 1849–1858. doi: 10.1007/s11517-017-1i632-z

Min, W., Yao, L., Lin, Z., and Liu, L. (2018). Support vector machine approach to fall recognition based on simplified expression of human skeleton action and fast detection of start key frame using torso angle. *IET Comput. Vis.* 12, 1133–1140. doi: 10.1049/iet-cvi.2018.5324

Namba, T., and Yamada, Y. (2018a). Fall risk reduction for the elderly by using mobile robots based on deep reinforcement learning. *J. Robot. Network. Artif. Life* 4, 265–269. doi: 10.2991/jrnal.2018.4.4.2

Namba, T., and Yamada, Y. (2018b). Risks of deep reinforcement learning applied to fall prevention assist by autonomous mobile robots in the hospital. *Big Data Cogn. Comput*. 2:13. doi: 10.3390/bdcc2020013

Niu, K., Zhang, F., Xiong, J., Li, X., Yi, E., and Zhang, D. (2018). "Boosting fine-grained activity sensing by embracing wireless multipath effects," in *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies* (Heraklion), 139–151. doi: 10.1145/3281411.3281425

Nukala, B., Shibuya, N., Rodriguez, A., Tsay, J., Nguyen, T., Zupancic, S., et al. (2014). "A real-time robust fall detection system using a wireless gait analysis sensor and an artificial neural network," in *2014 IEEE Healthcare Innovation Conference (HIC)* (Seattle: IEEE), 219–222. doi: 10.1109/HIC.2014.7038914

Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., and Bajcsy, R. (2013). "Berkeley MHAD: a comprehensive multimodal human action database," in *2013 IEEE Workshop on Applications of Computer Vision (WACV)* (Clearwater Beach, FL: IEEE), 53–60. doi: 10.1109/WACV.2013.6474999

Ozcan, K., and Velipasalar, S. (2016). Wearable camera-and accelerometer-based fall detection on portable devices. *IEEE Embed. Syst. Lett*. 8, 6–9. doi: 10.1109/LES.2015.2487241

Ozcan, K., Velipasalar, S., and Varshney, P. K. (2017). Autonomous fall detection with wearable cameras by using relative entropy distance measure. *IEEE Trans. Hum. Mach. Syst*. 47, 31–39. doi: 10.1109/THMS.2016.2620904

Palipana, S., Rojas, D., Agrawal, P., and Pesch, D. (2018). Falldefi: ubiquitous fall detection using commodity wi-fi devices. *Proc. ACM Interact. Mobile Wearable Ubiquit. Technol.* 1, 1–25. doi: 10.1145/3161183

Pandian, P., Mohanavelu, K., Safeer, K., Kotresh, T., Shakunthala, D., Gopal, P., et al. (2008). Smart vest: Wearable multi-parameter remote physiological monitoring system. *Med. Eng. Phys*. 30, 466–477. doi: 10.1016/j.medengphy.2007.05.014

Paradiso, R., Loriga, G., and Taccini, N. (2005). A wearable health care system based on knitted integrated sensors. *IEEE Trans. Inform. Technol. Biomed*. 9, 337–344. doi: 10.1109/TITB.2005.854512

Pierleoni, P., Belli, A., Palma, L., Pellegrini, M., Pernini, L., and Valenti, S. (2015). A high reliability wearable device for elderly fall detection. *IEEE Sens. J*. 15, 4544–4553. doi: 10.1109/JSEN.2015.2423562

Pister, K., Hohlt, B., Ieong, I., Doherty, L., and Vainio, I. (2003). *Ivy-a Sensor Network Infrastructure for the College of Engineering*. Available online at: http://www-bsac.eecs.berkeley.edu/projects/ivy

Putra, I., Brusey, J., Gaura, E., and Vesilo, R. (2017). An event-triggered machine learning approach for accelerometer-based fall detection. *Sensors* 18, 20. doi: 10.3390/s18010020

Queralta, J. P., Gia, T., Tenhunen, H., and Westerlund, T. (2019). "Edge-AI in Lora-based health monitoring: fall detection system with fog computing and LSTM recurrent neural networks," in *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)* (IEEE), 601–604. doi: 10.1109/TSP.2019.8768883

Ray, P. P. (2014). "Home health hub internet of things (H 3 IoT): an architectural framework for monitoring health of elderly people," in *2014 International Conference on Science Engineering and Management Research (ICSEMR)* (IEEE), 1–3. doi: 10.1109/ICSEMR.2014.7043542

Rougier, C., Auvinet, E., Rousseau, J., Mignotte, M., and Meunier, J. (2011a). "Fall detection from depth map video sequences," in *International Conference on Smart Homes and Health Telematics* (Montreal: Springer), 121–128. doi: 10.1007/978-3-642-21535-3_16

Rougier, C., Meunier, J., St-Arnaud, A., and Rousseau, J. (2011b). Robust video surveillance for fall detection based on human shape deformation. *IEEE Trans. Circ. Syst. Video Technol*. 21, 611–622. doi: 10.1109/TCSVT.2011.2129370

Sabatini, A. M., Ligorio, G., Mannini, A., Genovese, V., and Pinna, L. (2016). Prior-to-and post-impact fall detection using inertial and barometric altimeter measurements. *IEEE Trans. Neural Syst. Rehabil. Eng*. 24, 774–783. doi: 10.1109/TNSRE.2015.2460373

Saleh, M., and Jeannés, R. L. B. (2019). Elderly fall detection using wearable sensors: a low cost highly accurate algorithm. *IEEE Sens. J*. 19, 3156–3164. doi: 10.1109/JSEN.2019.2891128

Schwickert, L., Becker, C., Lindemann, U., Maréchal, C., Bourke, A., Chiari, L., et al. (2013). Fall detection with body-worn sensors. *Z. Gerontol. Geriatr*. 46, 706–719. doi: 10.1007/s00391-013-0559-8

Senouci, B., Charfi, I., Heyrman, B., Dubois, J., and Miteran, J. (2016). Fast prototyping of a SOC-based smart-camera: a real-time fall detection case study. *J. Real Time Image Process*. 12, 649–662. doi: 10.1007/s11554-014-0456-4

Shi, T., Sun, X., Xia, Z., Chen, L., and Liu, J. (2016). Fall detection algorithm based on triaxial accelerometer and magnetometer. *Eng. Lett*. 24:EL_24_2_06.

Shibuya, N., Nukala, B. T., Rodriguez, A., Tsay, J., Nguyen, T. Q., Zupancic, S., et al. (2015). "A real-time fall detection system using a wearable gait analysis sensor and a support vector machine (SVM) classifier," in *2015 Eighth International Conference on Mobile Computing and Ubiquitous Networking (ICMU)* (IEEE), 66–67. doi: 10.1109/ICMU.2015.7061032

Shojaei-Hashemi, A., Nasiopoulos, P., Little, J. J., and Pourazad, M. T. (2018). "Video-based human fall detection in smart homes using deep learning," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (Florence: IEEE), 1–5. doi: 10.1109/ISCAS.2018.8351648

Spasova, V., Iliev, I., and Petrova, G. (2016). Privacy preserving fall detection based on simple human silhouette extraction and a linear support vector machine. *Int. J. Bioautomat*. 20, 237–252.

Stone, E. E., and Skubic, M. (2015). Fall detection in homes of older adults using the Microsoft Kinect. *IEEE J. Biomed. Health Inform*. 19, 290–301. doi: 10.1109/JBHI.2014.2312180

Sucerquia, A., López, J., and Vargas-Bonilla, J. (2018). Real-life/real-time elderly fall detection with a triaxial accelerometer. *Sensors* 18:1101. doi: 10.3390/s18041101

Thilo, F. J., Hahn, S., Halfens, R. J., and Schols, J. M. (2019). Usability of a wearable fall detection prototype from the perspective of older people-a real field testing approach. *J. Clin. Nurs*. 28, 310–320. doi: 10.1111/jocn.14599

Tian, Y., Lee, G.-H., He, H., Hsu, C.-Y., and Katabi, D. (2018). RF-based fall monitoring using convolutional neural networks. *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol*. 2, 1–24. doi: 10.1145/3264947

Tsinganos, P., and Skodras, A. (2018). On the comparison of wearable sensor data fusion to a single sensor machine learning technique in fall detection. *Sensors* 18, 592. doi: 10.3390/s18020592

Wang, H., Zhang, D., Wang, Y., Ma, J., Wang, Y., and Li, S. (2017a). RT-fall: a real-time and contactless fall detection system with commodity wifi devices. *IEEE Trans. Mob. Comput*. 16, 511–526. doi: 10.1109/TMC.2016.2557795

Wang, Y., Wu, K., and Ni, L. M. (2017b). Wifall: device-free fall detection by wireless networks. *IEEE Trans. Mobile Comput*. 16, 581–594. doi: 10.1109/TMC.2016.2557792

WHO (2018). *Falls*. Available online at: https://www.who.int/news-room/fact-sheets/detail/falls

Williams, G., Doughty, K., Cameron, K., and Bradley, D. (1998). "A smart fall and activity monitor for telecare applications," in *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol. 20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No. 98CH36286)*, Volume 3 (IEEE), 1151–1154. doi: 10.1109/IEMBS.1998.747074

Wu, F., Zhao, H., Zhao, Y., and Zhong, H. (2015). Development of a wearable-sensor-based fall detection system. *Int. J. Telemed. Appl*. 2015:2. doi: 10.1155/2015/576364

Wu, T., Gu, Y., Chen, Y., Xiao, Y., and Wang, J. (2019). A mobile cloud collaboration fall detection system based on ensemble learning. *arXiv [Preprint]. arXiv:1907.04788*.

Xi, X., Jiang, W., Lü, Z., Miran, S. M., and Luo, Z.-Z. (2020). Daily activity monitoring and fall detection based on surface electromyography and plantar pressure. *Complexity*. 2020:9532067. doi: 10.1155/2020/9532067

Xi, X., Tang, M., Miran, S. M., and Luo, Z. (2017). Evaluation of feature extraction and recognition for activity monitoring and fall detection based on wearable SEMG sensors. *Sensors* 17, 1229. doi: 10.3390/s17061229

Xu, T., Zhou, Y., and Zhu, J. (2018). New advances and challenges of fall detection systems: a survey. *Appl. Sci*. 8, 418. doi: 10.3390/app8030418

Yang, G. (2018). *A Study on Autonomous Motion Planning of Mobile Robot by Use of Deep Reinforcement Learning for Fall Prevention in Hospita*. Japan: JUACEP Indenpedent Research Report Nagoya University.

Yang, G.-Z., and Yang, G. (2006). *Body Sensor Networks*. Springer. doi: 10.1007/1-84628-484-8

Yang, K., Ahn, C. R., Vuran, M. C., and Aria, S. S. (2016). Semi-supervised near-miss fall detection for ironworkers with a wearable inertial measurement unit. *Automat. Construct*. 68, 194–202. doi: 10.1016/j.autcon.2016.04.007

Yang, S.-W., and Lin, S.-K. (2014). Fall detection for multiple pedestrians using depth image processing technique. *Comput. Methods Programs Biomed*. 114, 172–182. doi: 10.1016/j.cmpb.2014.02.001

Yazar, A., Erden, F., and Cetin, A. E. (2014). "Multi-sensor ambient assisted living system for fall detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-14)* (Florence), 1–3.

Yun, Y., Innocenti, C., Nero, G., Lindén, H., and Gu, I. Y.-H. (2015). "Fall detection in RGB-D videos for elderly care," in *2015 17th International Conference on E-health Networking, Application* & *Services (HealthCom)* (Boston, MA: IEEE), 422–427.

Zhang, L., Wang, C., Ma, M., and Zhang, D. (2019). Widigr: direction-independent gait recognition system using commercial wi-fi devices. *IEEE Internet Things J*. 7, 1178–1191. doi: 10.1109/JIOT.2019.2953488

Zhang, T., Wang, J., Liu, P., and Hou, J. (2006). Fall detection by embedding an accelerometer in cellphone and using kfd algorithm. *Int. J. Comput. Sci. Netw. Security* 6, 277–284.

Zhang, Z., Conly, C., and Athitsos, V. (2014). "Evaluating depth-based computer vision methods for fall detection under occlusions," in *International Symposium on Visual Computing* (Las Vegas: Springer), 196–207. doi: 10.1007/978-3-319-14364-4_19

Zhang, Z., Conly, C., and Athitsos, V. (2015). "A survey on vision-based fall detection," in *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments* (Las Vegas: ACM), 46. doi: 10.1145/2769493.2769540

Zhao, M., Li, T., Abu Alsheikh, M., Tian, Y., Zhao, H., Torralba, A., et al. (2018). "Through-wall human pose estimation using radio signals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Long Beach, CA), 7356–7365. doi: 10.1109/CVPR.2018.00768

Zitouni, M., Pan, Q., Brulin, D., and Campo, E. (2019). Design of a smart sole with advanced fall detection algorithm. *J. Sensor Technol*. 9:71. doi: 10.4236/jst.2019.94007

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 28 August 2020
doi: 10.3389/frobt.2020.00106

Improving CT Image Tumor Segmentation Through Deep Supervision and Attentional Gates

Alžběta Turečková[1], Tomáš Tureček[1], Zuzana Komínková Oplatková[1] and Antonio Rodríguez-Sánchez[2]*

[1] *Artificial Intelligence Laboratory, Faculty of Applied Informatics, Tomas Bata University in Zlin, Zlin, Czechia,* [2] *Intelligent and Interactive Systems, Department of Computer Science, University of Innsbruck, Innsbruck, Austria*

Computer Tomography (CT) is an imaging procedure that combines many X-ray measurements taken from different angles. The segmentation of areas in the CT images provides a valuable aid to physicians and radiologists in order to better provide a patient diagnose. The CT scans of a body torso usually include different neighboring internal body organs. Deep learning has become the state-of-the-art in medical image segmentation. For such techniques, in order to perform a successful segmentation, it is of great importance that the network learns to focus on the organ of interest and surrounding structures and also that the network can detect target regions of different sizes. In this paper, we propose the extension of a popular deep learning methodology, Convolutional Neural Networks (CNN), by including deep supervision and attention gates. Our experimental evaluation shows that the inclusion of attention and deep supervision results in consistent improvement of the tumor prediction accuracy across the different datasets and training sizes while adding minimal computational overhead.

Keywords: medical image segmentation, CNN, UNet, VNet, attention gates, deep supervision, tumor segmentation, organ segmentation

OPEN ACCESS

Edited by:
George Azzopardi,
University of Groningen, Netherlands

Reviewed by:
Antonio Greco,
University of Salerno, Italy
Laura Fernáández-Robles,
Universidad de León, Spain

Correspondence:
Alžběta Turečková
tureckova@utb.cz

Specialty section:
This article was submitted to Sensor Fusion and Machine Perception, a section of the journal Frontiers in Robotics and AI

Received: *31 July 2019*
Accepted: *07 July 2020*
Published: *28 August 2020*

Citation:
Turečková A, Tureček T, Komínková Oplatková Z and Rodríguez-Sánchez A (2020) Improving CT Image Tumor Segmentation Through Deep Supervision and Attentional Gates. Front. Robot. AI 7:106. doi: 10.3389/frobt.2020.00106

1. INTRODUCTION

The daily work of a radiologist consists of visually analyzing multiple anatomical structures in medical images. Subtle variations in size, shape, or structure may be a sign of disease and can help to confirm or discard a particular diagnosis. However, manual measurements are time-consuming and could result in inter-operator and intra-operator variability (Sharma and Aggarwal, 2010; Jimenez-del-Toro et al., 2016). At the same time, the amount of data acquired via Computer tomography (CT) and Magnetic resonance (MR) is ever-growing (Sharma and Aggarwal, 2010). As a result, there is an increasing interest in reliable automatic systems that assist radiological experts in clinical diagnosis and treatment planning. One of such aids to experts is medical image segmentation, which consists of voxel-wise annotation of target structures in the image and it is present in many recent research work. Yearly medical image competition challenges[1] allow to the computer vision and machine learning experts to access and evaluate medical image data (Jimenez-del-Toro et al., 2016).

Deep learning techniques, especially convolutional neural networks (CNN), have become the state-of-the-art for medical image segmentation. Fully convolutional networks (FCNs)

[1]For example website Grand Challenges in Biomedical Image Analysis gathers multiple competitions; https://grand-challenge.org.

(Long et al., 2015) and the U-Net (Ronneberger et al., 2015) are two of the most commonly used architectures. Their area of application includes anatomical segmentation of cardiac CT (Zreik et al., 2016), detection of lung nodules in chest CT (Hamidian et al., 2017), multi-organ segmentation in CT and MRI images of the abdomen (Jimenez-del-Toro et al., 2016), and ischemic stroke lesion outcome prediction based on multispectral MRI (Winzeck et al., 2018) among others.

Despite the success of deep CNN techniques, there are difficulties inherent to their applicability. First, large datasets are needed for the successful training of deep CNN models. In medical imaging, this may be problematic due to the cost of acquisition, data anonymization techniques, etc. Second, volumetric medical image data require vast computational resources, even when using graphical computation units (GPU) the training process is very time-consuming. Therefore, every new proposal should take into account not only the performance but also the computational load.

Current CT-based clinical abdominal diagnosis relies on the comprehensive analysis of groups of organs, and the quantitative measures of volumes, shapes, and others, which are usually indicators of disorders. Computer-aided diagnosis and medical image analysis traditionally focus on organ or disease based applications, i.e., multi-organ segmentation from abdominal CT (Jimenez-del-Toro et al., 2016; Hu et al., 2017; Gibson et al., 2018), or tumor segmentation in the liver (Linguraru et al., 2012), the pancreas (Isensee et al., 2018), or the kidney (Yang et al., 2018).

There are two significant challenges in automatic abdominal organ segmentation from CT images (Hu et al., 2017). One of such challenges is how to automatically locate the anatomical structures in the target image because different organs lay close to each other and can also overlap. Moreover, among individual patients exists considerable variations in the location, shape, and size of organs. Furthermore, abdominal organs are characteristically represented by similar intensity voxels as identify surrounding tissues in CT images. The other challenge is to determine the fuzzy boundaries between neighboring organs and soft tissues surrounding them.

The task of detecting cancerous tissue in an abdominal organ is even more difficult because of the large variability of tumors in size, position, and morphology structure. Results are quite impressive when the focus is on detecting organs; an example of this is (Isensee et al., 2018), achieving dice scores of 95.43 and 79.30 for liver and pancreas segmentation. On the other hand, these values drop dramatically when the focus is on detecting the tumor, where values are as low as 61.82 and 52.12 for their respective (liver and pancreas) tumor classes. There is also a high variability on tumor classification depending on the organ, e.g., Yang et al. (2018) presents dice scores of 93.1 and 80.2 when the organ is the kidney and its tumor detection, respectively.

On the other hand, all the organs have a typical shape, structure, and relative position in the abdomen. The model could then benefit from an attentional mechanism consolidated in the network architecture, which could help to focus specifically on the organ of interest. For this purpose, we incorporated the idea of attention gates (AG) (Oktay et al., 2018). Attention gates identify salient image regions and prune feature responses to preserve only the activations relevant to the specific task and to suppress feature responses in irrelevant background regions without the requirement to crop the region of interest.

Many research papers have incorporated attention into artificial CNN visual models for image captioning (Xu et al., 2015), classification (Mnih et al., 2014; Xiao et al., 2015), and segmentation (Chen et al., 2016). In the case of Recurrent Neural Networks (RNN), Ypsilantis and Montana (2017) presents an RNN model that learns to sequentially sample the entire X-ray image and focus only on salient areas. In these models, attention could be divided into two categories: hard and soft attention. As described by Xu et al. (2015), hard attention is when the attention scores are used to select a single hidden state, e.g., iterative region proposal and cropping. Such an attention mechanism is often non-differentiable and relies on reinforcement learning for updating parameter values, which makes training quite challenging. On the other hand, soft attention calculates the context vector as a weighted sum of the encoder hidden states (feature vectors). Thus, soft attention is differentiable, and the entire model is trainable by back-propagation. The attention modules which generate attention-aware features presented by Wang et al. (2017) was the state-of-the-art object recognition performance on ImageNet in 2017. Huang et al. (2019) presents a Criss-Cross Network (CCNet) with a criss-cross attention module and achieves the state-of-the-art results of mIoU score of 81.4 and 45.22 on Cityscapes test set and ADE20K validation set, respectively. Grewal et al. (2018) combines deep CNN architecture with the components of attention for slice level predictions and achieves 81.82% accuracy for the prediction of hemorrhage from 3D CT scans, matching the performance of a human radiologist. Other boosted convolutional neural network with attention and deep supervision (DAB-CNN) (Kearney et al., 2019) achieves state-of-the-art results in automatic segmentation of the prostate, rectum, and penile bulb.

Deep supervision was firstly introduced by Lee et al. (2015) as a way to deal with the problem of the vanishing gradient in training deeper CNN for image classification. This method adds companion objective functions at each hidden layer in addition to the overall objective function at the output layer. Such a model can learn robust features even in the early layers; moreover the deep supervision brings some insight on the effect that intermediate layers may have on the overall model performance. Since then, deep supervision was successfully applied in many vision models. In the case of medical applications, it has been employed to prostate segmentation (Zhu et al., 2017), to the liver (Dou et al., 2016), and pancreatic cyst (Zhou et al., 2017) segmentation in CT volumes, and to brain tumor segmentation from magnetic resonance imaging (Isensee et al., 2017).

In the present work, we propose a methodology for a more reliable organ and tumor segmentation from computed tomography scans. The contribution of this work is three-fold:

- A methodology that achieves the state-of-the-art performance on several segmentation tasks dealing with organ and tumor

segmentation, of special interest is the increase obtained in the precision of tumor segmentation.
- A visualization of the feature maps from our CNN architecture to provide some insight into what is the focus of attention in the different parts of the model for better tumor detection.
- Third and not last, we provide a novel and extended comparison of CNN architectures for different organ-tumor segmentation from abdomen CT scans.

2. METHODOLOGY

We will provide the details of the proposed methodology in this section. Firstly, we will explain the preprocessing and normalization of the medical image data. Secondly, we will provide a detailed description of the model architecture, the attention gates, and the deep supervision layers. The loss function, the optimizer, and other specifics of interest are detailed in the following subsection, which also describes patch sampling and data augmentation techniques utilized in order to prevent overfitting. The last part shortly outlines inference and how the image patches are stitched back together. We provide a publicly available implementation of our methodology using PyTorch at: github.com/tureckova/Abdomen-CT-Image-Segmentation.

2.1. Data Preprocessing

CT scans might be captured by different scanners in different medical clinics with nonidentical acquisition protocols; therefore the data preprocessing step is crucial to normalize the data in a way that enables the convolutional network to learn suitable and meaningful features properly. We preprocess the CT scan images as follows (Isensee et al., 2018):

- All patients are resampled to the median voxel spacing of the dataset using the third-order spline interpolation for image data and the nearest neighbor interpolation for the segmentation mask.
- The dataset is normalized by clipping to the [0.5, 99.5] percentiles of the intensity values occurring within the segmentation masks.
- Z-score normalization is applied based on the mean and standard deviation of all intensity values occurring within the segmentation masks.

Because of memory restrictions, the model was trained on 3D image patches. All the models were trained on an 11GB GPU. A base configuration of the input patch size of 128 × 128 × 128 and a batch size of 2 was chosen to fit our hardware set up. Then the model automatically adapts these parameters, so they reflect the median image size of each dataset. We consider two different approaches:

Full-resolution—the original resolutions of images are used for the training, and relatively small 3D patches are chosen randomly during training. This way, the network has access to high-resolution details; on the other hand, it neglects context information.

Low-resolution—the patient image is downsampled by a factor of two until the median shape of the resampled data has less than four times the voxels that can be processed as an input patch. 3D patches are also chosen randomly during training. In this case, the model has more information about the context but lacks high-resolution details.

2.2. Model Architecture

Deep learning techniques, especially convolutional neural networks, occupy the main interest of research in the area of medical image segmentation nowadays and outperform most techniques. A very popular convolution neural network architecture used in medical imaging is the encoder-decoder structure with skip connections at each image resolution level. The basic principle was firstly presented by Ronneberger et al. (2015) for segmenting 2D biomedical images; this network was named U-Net. U-Net traditionally uses the max-pooling to downsample the image in the encoder part and upsampling in the decoder part of the structure. The work of Milletari et al. (2016) extended the model for volumetric medical image segmentation and replaced the max-pooling and upsampling by convolutions, creating a fully convolutional neural network named V-Net. The original U-Net architecture was quickly extended into 3D, and since then, the literature seems to be using names U-Net and V-Net interchangeably. In this work, all models work with volumetric data, and we decided to keep the original architectures naming and differences:

- **UNet**—the encoder-decoder structure with the skip connections using the max-pooling to downsample the image in the encoder part and upsampling in the decoder part of the structure.
- **VNet**—the fully convolutional encoder-decoder architecture with skip connections.

We follow encoder-decoder architecture choices applied to each dataset by Isensee et al. (2018). We use 30 feature maps in the highest layers (the number of feature maps doubles with each downsampling), and we downsample the image along each axis until the feature maps have size 8 or for a maximum of 5 times. The encoder part consists of context modules, and the decoder part is composed of localization modules. Each module contains a convolution layer, a dropout layer, an instance normalization layer, and a leakyReLU.

In addition to original encoder-decoder network architecture, we add attention gates (Oktay et al., 2018) in the top two model levels and deep supervision (Kayalibay et al., 2017). Both extensions are described in the next two subsections. The structure of proposed network architecture is shown in **Figure 1**.

2.2.1. Attention Gates

Attention coefficients, $\alpha_i \in [0, 1]$ emphasizes salient image regions and significant features to preserve only relevant activations specific to the actual task. The output of AGs (1) is the element-wise multiplication of input feature-maps and attention coefficients:

$$\hat{x}^l_{i,c} = x^l_{i,c} \cdot \alpha^l_{i,c} \tag{1}$$

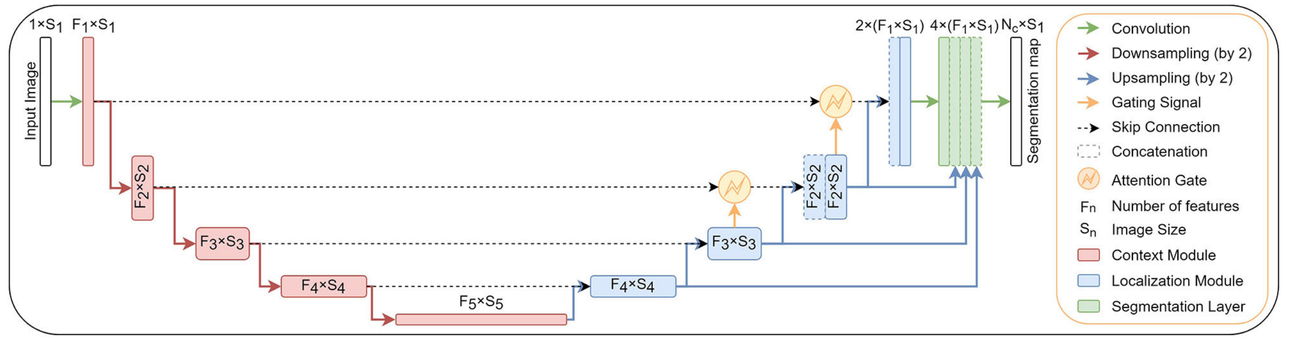

FIGURE 1 | A block diagram of the segmentation model with attention gates and deep supervision.

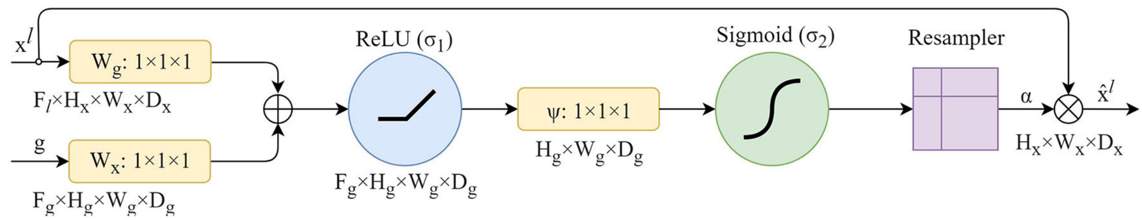

FIGURE 2 | A block diagram of additive attention gate (AG) (Oktay et al., 2018). Input features (x^l) are scaled with the attention coefficients (α) computed in AG. Spatial regions are selected by analyzing both the activations and the contextual information provided by the gating signal (g) which is collected from a coarser resolution scale. Attention coefficients are resampled to match the resolution of (x^l) by trilinear interpolation.

where $\alpha^l_{i,c}$ is the attention coefficient (obtained using Equation 3, below), and $x^l_{i,c}$ is pixel i in layer l for class c. $x^l_i \in \mathbb{R}^{F_l}$ where F_l corresponds to the number of feature-maps in layer l. Therefore, each AG learns to focus on a subset of target structures. The structure of an attention gate is shown in **Figure 2**. A gating vector g_i is used for each pixel i to determine the regions of focus. The gating vector contains contextual information to reduce lower-level feature responses. The gate uses additive attention (2), formulated as follows (Oktay et al., 2018):

$$q^l_{att} = \psi^T(\sigma_1(W^T_x x^l_{i,c} + W^T_g g_{i,c} + b_g)) + b_\psi \quad (2)$$

$$\alpha^l_{i,c} = \sigma_2(q^l_{att}(x^l_{i,c}, g_{i,c}, \Theta_{att})), \quad (3)$$

where $\sigma_1(x^l_{i,c}) = max(0, x^l_{i,c})$ is rectified linear unit. AG is characterized by a set of parameters Θ_{att} containing: linear transformations $W_x \in \mathbb{R}^{F_l \times F_{int}}$, $W_g \in \mathbb{R}^{F_g \times F_{int}}$, $\psi \in \mathbb{R}^{F_{int} \times 1}$ and bias terms $b_\psi \in \mathbb{R}$, $b_g \in \mathbb{R}^{F_{int}}$. $\sigma_2(x^l_{i,c}) = \frac{1}{1+exp(-x^l_{i,c})}$ corresponds to a sigmoid activation function. The linear transformations are computed using channel-wise $1 \times 1 \times 1$ convolutions of the input tensors. All the AG parameters can be trained with the standard back-propagation updates.

2.2.2. Deep Supervision

Deep supervision (Kayalibay et al., 2017) is the design where multiple segmentation maps are generated at different resolutions levels. The feature maps from each network level are transposed by $1 \times 1 \times 1$ convolutions to create secondary segmentation maps. These are then combined in the following way: First, the segmentation map with the lowest resolution is upsampled with bilinear interpolation to have the same size as the second-lowest resolution segmentation map. The element-wise sum of the two maps is then upsampled and added to the third-lowes segmentation map and so on until we reach the highest resolution level. For illustration see **Figure 1**.

These additional segmentation maps do not primarily serve for any further refinement of the final segmentation map created at the last layer of the model because the context information is already provided by long skip connections. The secondary segmentation maps help in the speed of convergence by "encouraging" earlier layers of the network to produce better segmentation results. A similar principle has been used by Kayalibay et al. (2017) and Chen et al. (2018).

2.3. Training

Unless stated otherwise, all models are trained with a five-fold cross-validation. The network is trained with a combination of dice (5) and cross-entropy (6) loss function (4):

$$L_{total} = L_{dice} + L_{crossEntropy}, \quad (4)$$

$$L_{dice} = -\frac{2}{|C|} \sum_{c \in C} \frac{\sum_{i \in I} u^c_i v^k_i}{\sum_{i \in I} u^c_i + \sum_{i \in I} v^c_i}, \quad (5)$$

$$L_{crossEntropy} = -\sum_{c \in C} \sum_{i \in I} (v_i^c \log(u_i^k)), \quad (6)$$

where u is the softmax output of the network and v is a one hot encoding of the ground truth segmentation map[2]. Both u and v have shape $I \times C$ with $i \in I$ being the number of pixels in the training patch/batch and $c \in C$ being the classes. The cross-entropy loss speeds up the learning in the beginning of the training, while the dice loss function helps to deal with the label unbalance which is typical for medical images data.

The dice loss is computed for each class and each sample in the batch and averaged over the batch and over all classes. We use the Adam optimizer with an initial learning rate 3×10^{-5} and l_2 weight decay 3×10^{-5} for all experiments. An epoch is defined as the iteration over all training images. Whenever the exponential moving average of the training loss does not improve within the last 30 epochs, the learning rate is decreased by a factor of 0.2. We train till the learning rate drops below 10^{-6} or $1,000$ epochs are exceeded.

Gradient updates are computed by standard backpropagation using a small batch size of 2. Initial weights values are extracted from a normal distribution (He et al., 2015). Gating parameters are initialized such that the attention gates let pass all feature vectors at all spatial locations.

2.3.1. Data Augmentation and Patch Sampling

Training of the deep convolutional neural networks from limited training data suffers from overfitting. To minimize this problem, we apply a large variety of data augmentation techniques: random rotations, random scaling, random elastic deformations, gamma correction augmentation, and mirroring. All the augmentation techniques are applied on the fly during training. Data augmentation is realized with a framework which is publicly available at: https://github.com/MIC-DKFZ/batchgenerators.

The patches are generated randomly on the fly during the training, but we force that minimally one of the samples in a batch contains at least one foreground class to enhance the stability of the network training.

2.4. Inference

According to the training, inference of the final segmentation mask is also made patch-wise. The output accuracy is known to decrease toward the borders of the predicted image. Therefore, we overlap the patches by half the size of the patch and also weigh voxels close to the center higher than those close to the border, when aggregating predictions across patches. The weights are generated, so the center position in a patch is equal to one, and the boundary pixels are set to zero, in between the values are extracted from a Gaussian distribution with sigma equal to one-eight of patch size. To further increase the stability, we use test time data augmentation by mirroring all patches along all axes.

[2] A one-hot encoding was created from the original ground true segmentation map in a way, that each image channel contains only one class present in segmentation map, this way all the classes are represented by value one just in different image channels. For example, if we have ground true segmentation map of size ($1 \times imSize1 \times imSize2 \times imSize3$) with three labels: 0, 1, 2. The one-hot encoding would have the size ($3 \times imSize1 \times imSize2 \times imSize3$).

TABLE 1 | An overview of image shapes, training setups, and network topologies for each task.

		High resolution	**Low resolution**
Kidney	Num. images training	168	168
	Num. images validation	42	42
	Median patient shape	511 × 511 × 136	247 × 247 × 127
	Input patch size	160 × 160 × 48	128 × 128 × 80
	Num. downsampling per axis	5, 5, 3	5, 5, 4
	Batch size	2	2
Liver	Num. images training	105	105
	Num. images validation	26	26
	Median patient shape	482 × 512 × 512	189 × 201 × 201
	Input patch size	96 × 128 × 128	96 × 128 × 128
	Num. downsampling per axis	5, 5, 5	5, 5, 5
	Batch size	2	2
Pancreas	Num. images training	224	224
	Num. images validation	57	57
	Median patient shape	96 × 512 × 512	88 × 299 × 299
	Input patch size	40 × 192 × 160	64 × 128 × 128
	Num. downsampling per axis	3, 5, 5	3, 5, 5
	Batch size	2	2

3. EXPERIMENTAL EVALUATION AND DISCUSSION

In order to show the validity of the proposed segmentation method, we evaluate the methodology on challenging abdominal CT segmentation problem. We appraise the detection of cancerous tissue inside three different organs: pancreas, liver, and kidney.

3.1. CT Scan Datasets

The experiments are evaluated on three different CT abdominal datasets featuring organ and tumor segmentation classes: kidney, liver, and pancreas. Each dataset brings slightly different challenges for the model. More information about each task dataset, training setups, and concrete network topologies are as follows (see also **Table 1**).

3.1.1. Kidney

The dataset features a collection of multi-phase CT imaging, segmentation masks, and comprehensive clinical outcomes for 300 patients who underwent nephrectomy for kidney tumors at the University of Minnesota Medical Center between 2010 and 2018 (Heller et al., 2019). Seventy percent (210) of these patients have been selected at random as the training set for the 2019 MICCAI KiTS Kidney Tumor Segmentation Challenge[3] and have been released publicly.

We perform five-fold cross-validation during training: 42 images are used for validation and 168 images for training. The mean patient shape after the resampling is 511 × 511 × 136 pixels

[3] kits19.grand-challenge.org

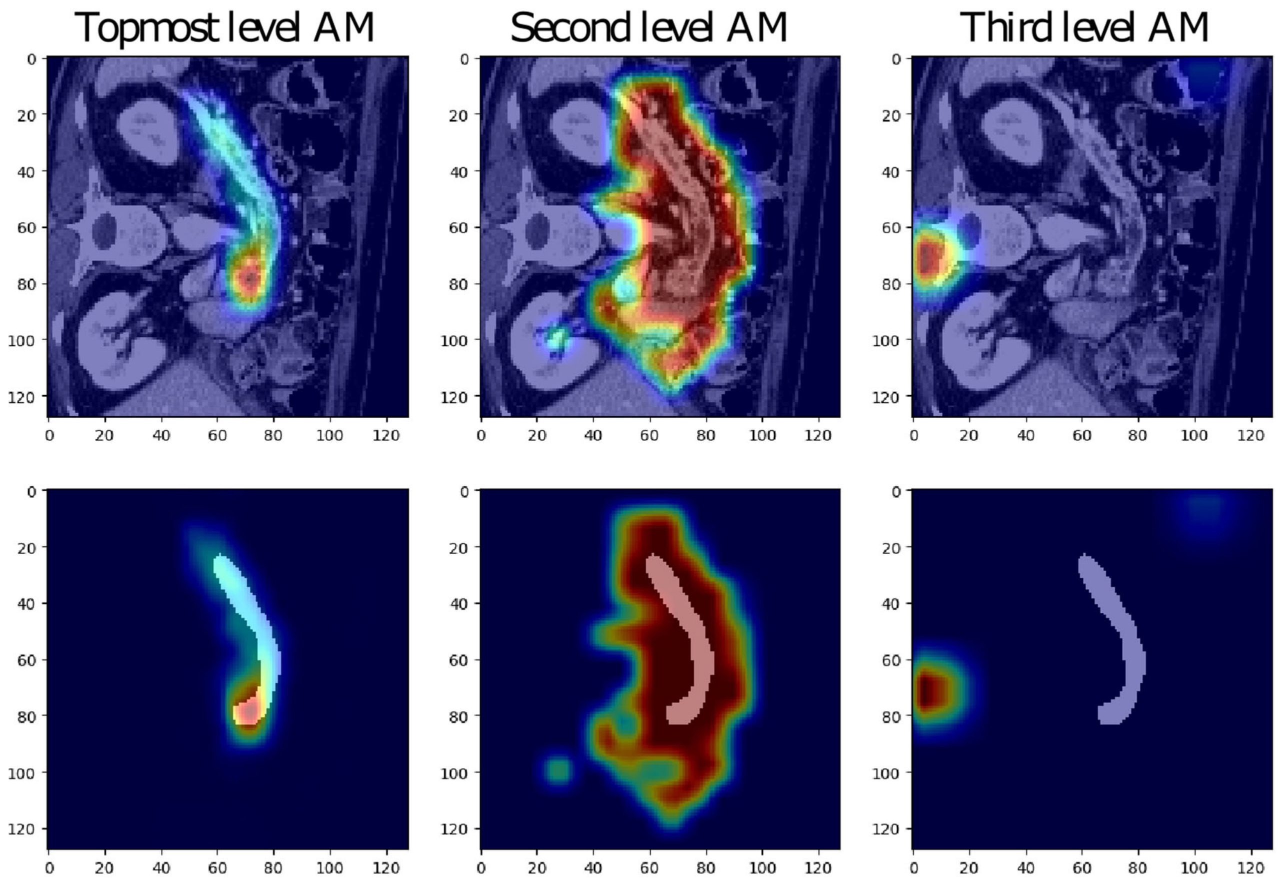

FIGURE 3 | Examples of attention maps (AM) obtained from attention gates in the three topmost levels of the low-resolution VNet (from left to right: full spatial resolution, downsampling of two and four).

in case of high-resolution and 247 × 247 × 127 pixels in low-resolution. According to the median shapes, we use 5, 5, and 3 downsampling for each respective image axis in high-resolution and 5, 5, 4 downsamplings in low-resolution. The patch size in case of high-resolution is $160 \times 160 \times 48$ pixels and $128 \times 128 \times 80$ pixels for low-resolution.

3.1.2. Liver

The dataset features a collection of 201 portal-venous-phase CT scans and segmentation masks for liver and tumor captured at IRCAD Hôpitaux Universitaires. Sixty-five percent (131) of these images have been released publicly as the training set for the 2018 MICCAI Medical Decathlon Challenge[4] (Simpson et al., 2019). This dataset contains a big label unbalance between organ (liver) and tumor. The inclusion of the dice term in the loss function (section 2.3) helps to mitigate the negative effects of such unbalance.

We perform five-fold cross-validation during training: 26 images are used for validation and 105 images for training. The mean patient shape after the resampling is 482 × 512 × 512 pixels in case of high-resolution and 189 × 201 × 201 pixels in low-resolution. According to the median shapes, we downsample five times each respective image axis in both high-resolution and low-resolution. The patch size in case of high-resolution was $96 \times 128 \times 128$ pixels and $96 \times 128 \times 128$ pixels for low-resolution.

3.1.3. Pancreas

The dataset features a collection of 421 portal-venous-phase CT imaging and segmentation masks for pancreas and tumor captured at Memorial Sloan Kettering Cancer Center. Seventy percent (282) of these images have been released publicly as the training set for the 2018 MICCAI Medical Decathlon Challenge[4] (Simpson et al., 2019). This dataset is also class unbalanced, the background being the most prominent class, followed by the organ (pancreas) and the tumor as the least present class. Appearance is quite heterogeneous for pancreas and tumor. As before, the inclusion of the dice term in the loss function helps to mitigate the negative effects of such unbalance.

We perform five-fold cross-validation during training: 26 images are used for validation and 105 images for training. The mean patient shape after the resampling is 96 × 512 × 512 pixels in the case of high-resolution and 88 × 299 × 299 pixels in low-resolution. According to the median shapes, we do 3, 5, and 5 downsampling for each respective image axis in high-resolution

[4]medicaldecathlon.com

and 3, 5, 5 downsamplings in low-resolution. The patch size in case of high-resolution is $40 \times 192 \times 160$ pixels and $64 \times 128 \times 128$ pixels for low-resolution.

3.2. Visualization of the Activation Maps

The network design allows us to visualize meaningful activations maps from the attention gates as well as from the deep supervision layers. The visualizations enable an exciting insight into the functionality of the convolutional network. The understanding of how the model represents the input image at the intermediate layers can help to gain more insight into improving the model and uncover at least part of the black-box behavior for which the neural networks are also known.

3.2.1. Visualization of the Attentional Maps

The low-resolution VNet was chosen to study the attention coefficients generated at different levels of a network trained on the Medical Decathlon Pancreas dataset. **Figure 3** shows the attention coefficients obtained from three top network levels (working with full spatial resolution and downsampled two and four times). The attention gates provide a rough outline of the organs in top two network levels, but not in the lower spatial resolution cases. For this reason, in our experiments, we decided to implement the AG only at two topmost levels and save the computation memory to handle larger image patches.

The attention coefficients obtained from two randomly chosen validation images from each studied dataset are visualized in **Figure 4**. All visualized attention maps correlate with the organ of interest, which indicates that the attention mechanism is focusing on the areas of interest, i.e., it emphasizes the salient image regions and significant features relevant for organ segmentation. In the case of liver segmentation, the attention map correlates accurately with the organ on the second level while in the top-level, the attention seems to focus on the organ borders. In kidney and pancreas datasets, we can observe exactly the opposite behavior. The attention map from top-level covers the organ, and the second level attention map focuses on the borders and the close organ surroundings. This difference is possibly associated with the different target sizes as the liver is taking a substantially larger part of the image than the kidney or pancreas.

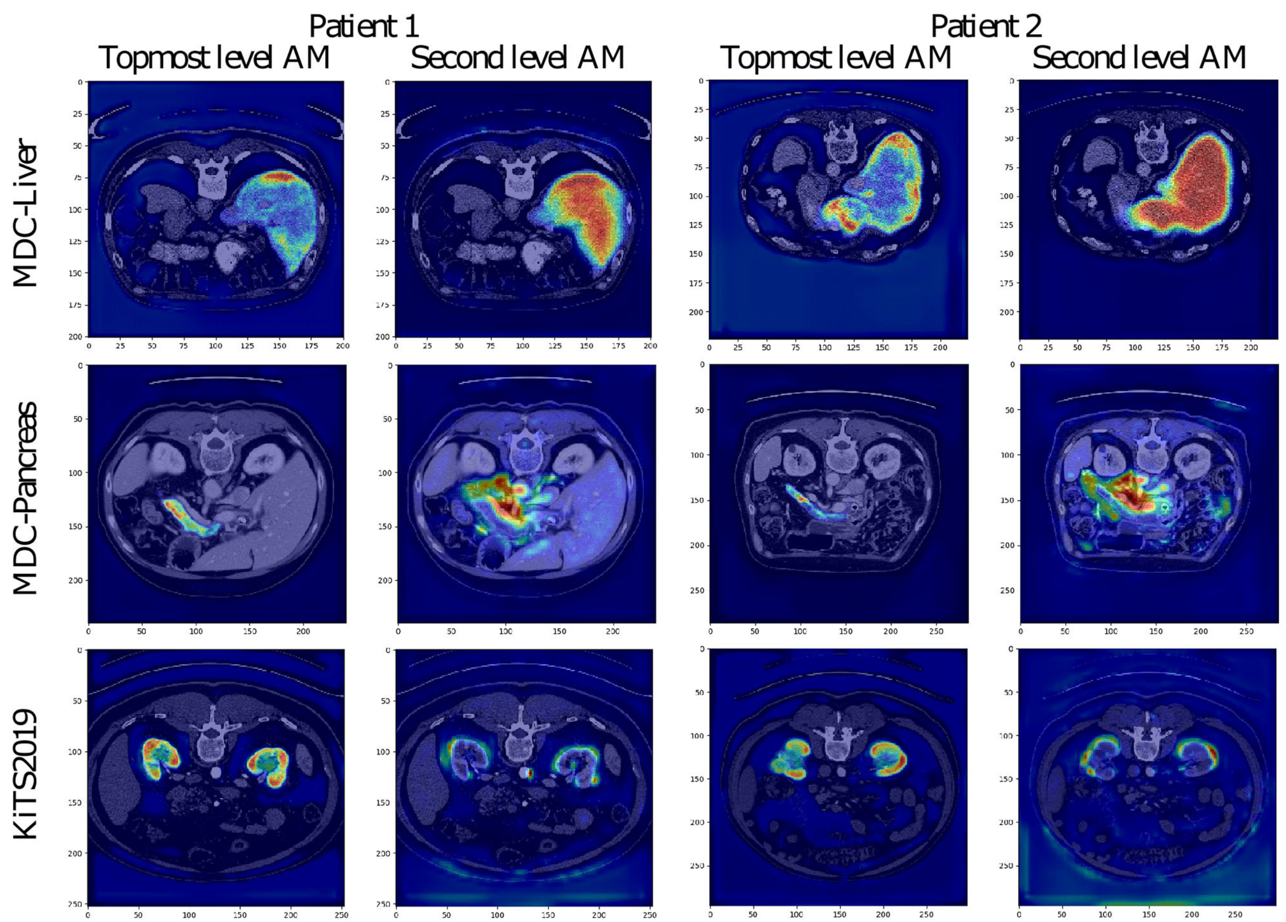

FIGURE 4 | Visualization of attention maps (AM) in low-resolution for VNet and two randomly chosen patient images from the validation set of each studied dataset. For each patient, the left picture shows the attention from the topmost layer (with the highest spatial resolution), and the right picture shows the attention from the second topmost layer.

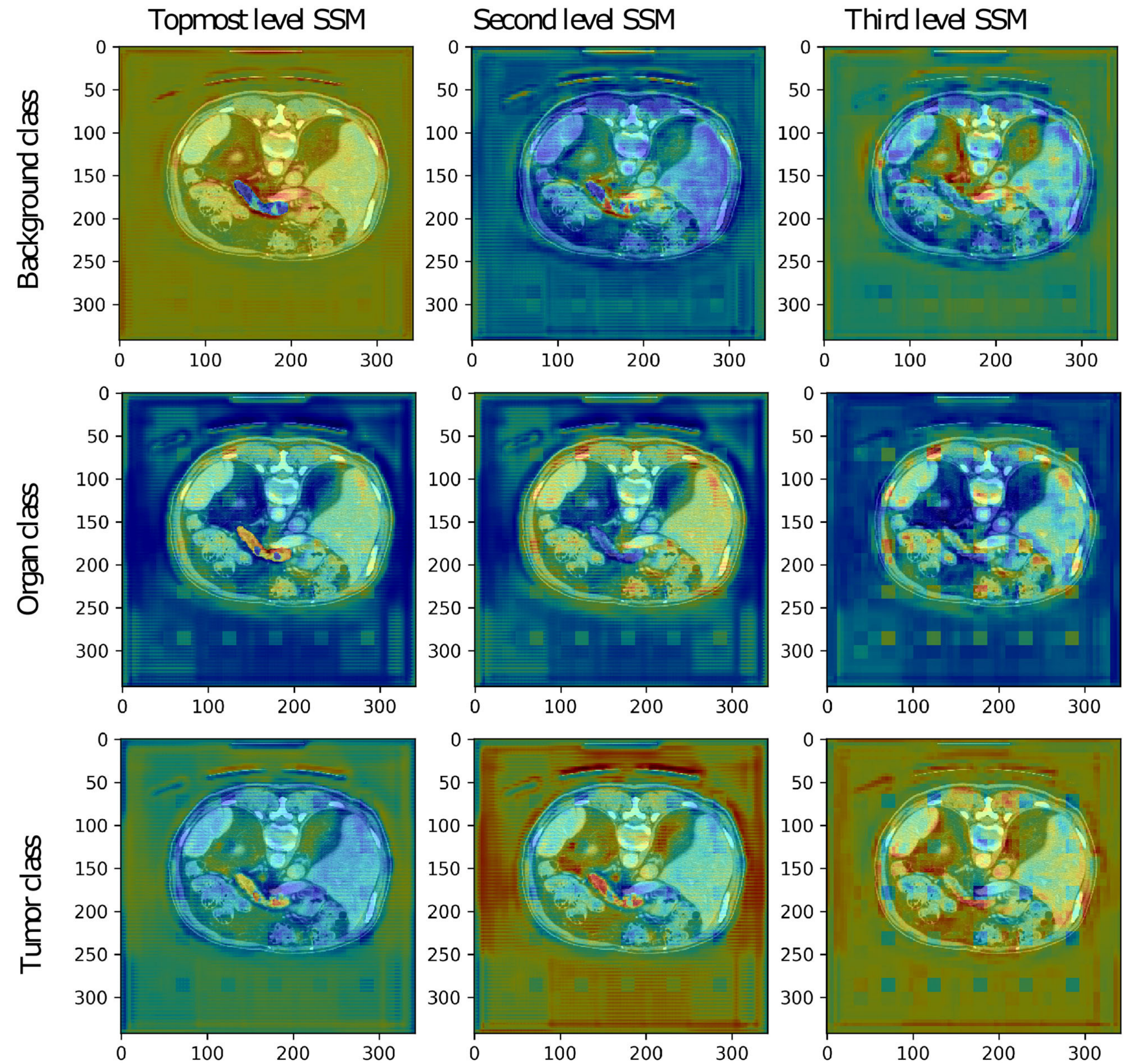

FIGURE 5 | The secondary segmentation maps (SSM) obtained from deep supervision layers of low-resolution VNet for one randomly chosen patient image from the validation set of the Medical Decathlon Pancreas dataset.

3.2.2. Visualization of the Deep Supervision Segmentation Maps

The low-resolution VNet was also chosen to study the secondary segmentation maps created at lower levels of the network trained on the Medical Decathlon Pancreas dataset. The segmentation maps are shown in **Figure 5**. Although the primary aim of the secondary segmentation maps is not the refinement of the final segmentation created at the last layer of the model, we could see the correlation between the occurrence of each label and the activation in the segmentation maps. The topmost segmentation map copies the final output. The second and third levels of activation are noisier, as it would be expected. We could see higher activations around the pancreas in the tumor class channels and also higher activations around the borders of the organ in the background label channel.

The more in-depth segmentation maps in the organ label channel are more challenging to interpret. The second level map seems to be inverted, including the pancreas into a darker part of the input image. On the other hand, the third level map highlights all the organs present in the image. After a summation of these two maps, we achieve the desired highlight of the pancreas. Overall, we could say that all the secondary segmentation maps have a relevant impact on the final result.

3.3. Evaluation Metrics

We use the following metrics score to evaluate the final segmentation in the subsequent sections: precision, recall, and dice. Each of the metrics is briefly explained below.

In the context of segmentation, precision, and recall compare the results of the classifier under test with the ground-true segmentation by a combination the true positives (T_P), true negatives (T_N), false positives (F_P), and false negatives (F_N). The terms positive and negative refer to the classifier's prediction, and the terms true and false refer to whether that prediction corresponds to the ground-truth labels. To summarize, Precision P (7) and Recall R (8) are determined as follows:

$$P = \frac{T_P}{T_P + F_P} * 100, \tag{7}$$

$$R = \frac{T_P}{T_P + F_N} * 100. \tag{8}$$

This way both the precision and recall are normalized in the range $\langle 0, 100 \rangle$, higher values indicating better performance.

When applied to a binary segmentation task, the dice score evaluates the degree of overlap between the predicted segmentation mask and the reference segmentation mask. Given binary masks, U and V, the Dice score D (9) is defined as:

$$D = \frac{2 * |U \cup V|}{|U| + |V|} * 100. \tag{9}$$

In this variant, the dice score lays in the range $\langle 0, 100 \rangle$, higher values indicating better performance.

3.4. Evaluating Four Architectures and Three Datasets

Next, we present a comprehensive study of the organ and tumor segmentation tasks on the three different abdominal CT datasets. For each dataset, four model variants were trained to show the impact of the different model architecture choices. The UNet utilizes max-pooling and the upsampling layers, while VNet is fully convolutional. Each architecture variant was trained on two different image resolutions: full-resolution and low-resolution. For more details about the model variants, please refer to section 2.2. Moreover, we provide assembly results from the respective full and low-resolution models. The soft-max output maps from the full and the low-resolution model variant were averaged and only then the final segmentation map was created. **Tables 2–4** summarize the results from five-fold cross-validation for all model variants for the Medical Decathlon Challenge (MDC) Liver dataset, the Medical Decathlon Challenge Pancreas dataset and the Kidney Tumor Segmentation Challenge (KiTS) dataset, respectively.

Due to the prominent inter-variability of position, size, and morphology structure, the tumor labels segmentation was less successful than the organ segmentation. We can see lower score values and also more significant inter-variability between the folds. The variability is especially high in the Liver-tumor label, where the lesions are usually divided into many small occurrences, and missing some of them means a significant change in the segmentation score results. The model could benefit from some postprocessing, which may help to sort out some of the lesions outside the liver organ, as suggested in Bilic et al. (2019). The overall scores are the lowest for the MDC Pancreas dataset. The variability in shape and size of the pancreas makes its segmentation a challenging task. Nevertheless, the attention mechanism helps the network to find the pancreas, thus obtaining a reasonably good performance.

Generally, the performance of the UNet and the fully convolutional VNet is comparable, but we could observe slightly better scores achieved by VNet in the MDC Liver dataset and KiTS dataset while the trend is opposed in the MDC Pancreas dataset, where the UNet provided better results than the VNet. Still, when it comes to the assembly results, the VNet benefits from its trainable parameters and achieves better results than UNet variant in all three datasets.

3.5. Performance Comparison

The proposed network architecture was benchmarked against the winning submission of the Medical Decathlon Challenge (MDC), namely nnUNet (Isensee et al., 2018) on two tasks: Task03-Liver and Task07-Pancreas. **Table 5** shows the mean dice scores from five-fold cross-validation for the low and the full-resolution variants of models as well as the best model presented in either work. The winning results from nnUNet consist of the combined prediction from three different models (2D UNet, 3D UNet, and 3D UNet cascade) assembled together. Therefore, we have chosen to compare also the results from 3D UNet model, whose model architecture is close to our network to highlight the difference gained by the network architecture changes, namely attention gates and deep supervision.

The full- and the low-resolution models with attention gates (VNet-AG-DSV) achieved higher dice scores for both labels on the pancreas dataset, of particular interest is that the tumor dice scores were substantially increased, by three and seven points in low and full-resolution, respectively. In the case of the liver dataset, we could see a significant improvement in the low-resolution case. Attention gates improved the tumor dice score by seven points while the liver segmentation precision was comparable. The decrease in dice score happened only on the tumor class in the full-resolution case. Finally, if we compare the best models presented in both papers, our model with attention gates and deep supervision (VNet-AG-DSV) wins on both datasets, adding nearly three score points on the liver-tumor class and two points in pancreas label.

The performance of the model with and without the attention gates is quantitatively compared in **Table 6**. We could see that both the number of parameters and the training and evaluation time increased just slightly, while the performance improvement was considerable. We should mention that the decrease in the number of parameters in the work of Isensee et al. (2018) was compensated by training the network with larger patch size: $128 \times 128 \times 128$ pixels versus $96 \times 128 \times 128$ pixels for the Liver

TABLE 2 | Kidney Tumor Challenge 2019.

Architecture		Kidney label			Tumor label		
		Precision	Recall	Dice	Precision	Recall	Dice
UNet	Low Res.	94.96 ± 0.02	96.22 ± 0.08	95.50 ± 0.01	81.51 ± 2.30	82.62 ± 3.85	79.27 ± 0.30
	Full res.	95.55 ± 0.75	97.08 ± 1.21	96.21 ± 0.62	78.83 ± 5.21	81.44 ± 4.63	76.70 ± 2.46
	Assembly	96.22 ± 1.32	97.11 ± 1.87	96.25 ± 1.12	83.88 ± 3.01	81.50 ± 6.23	78.68 ± 5.93
VNet	Low res.	94.79 ± 0.78	95.07 ± 1.42	94.63 ± 0.88	77.85 ± 3.43	78.51 ± 2.79	74.12 ± 2.66
	Full res.	96.01 ± 0.71	96.15 ± 1.19	95.93 ± 0.54	78.77 ± 3.60	79.72 ± 2.57	75.43 ± 1.59
	Assembly	96.54 ± 1.06	96.63 ± 1.35	96.43 ± 1.06	82.71 ± 2.80	83.39 ± 8.21	79.94 ± 5.33

Metrics scores from five-fold cross validation.

TABLE 3 | Medical Decathlon Challenge 2018—Task03-Liver.

Architecture		Liver label			Tumor label		
		Precision	Recall	Dice	Precision	Recall	Dice
UNet	Low res.	95.01 ± 0.92	95.52 ± 1.38	94.91 ± 1.57	63.65 ± 4.92	58.13 ± 7.66	53.27 ± 4.57
	Full res.	95.39 ± 1.03	96.28 ± 1.09	95.80 ± 1.16	58.24 ± 7.23	76.39 ± 9.51	58.87 ± 3.01
	Assembly	95.95 ± 0.70	96.66 ± 1.68	96.28 ± 1.01	63.74 ± 9.51	72.86 ± 10.1	60.29 ± 3.85
VNet	Low res.	94.96 ± 0.87	95.19 ± 1.75	94.54 ± 1.97	65.17 ± 5.69	59.13 ± 11.5	54.72 ± 6.11
	Full res.	94.39 ± 1.23	95.59 ± 1.03	94.86 ± 1.25	61.12 ± 8.33	70.34 ± 9.36	57.74 ± 2.20
	Assembly	95.57 ± 0.65	95.80 ± 1.36	95.74 ± 0.89	73.42 ± 5.76	67.41 ± 13.0	64.70 ± 3.08

Metrics scores from five-fold cross validation.

TABLE 4 | Medical Decathlon Challenge 2018—Task07-Pancreas.

Architecture		Pancreas label			Tumor label		
		Precision	Recall	Dice	Precision	Recall	Dice
UNet	Low res.	80.39 ± 1.83	83.70 ± 2.02	80.96 ± 2.33	62.18 ± 3.35	58.12 ± 6.12	54.66 ± 4.54
	Full res.	80.88 ± 1.66	83.77 ± 0.59	81.15 ± 0.43	60.86 ± 1.41	54.36 ± 3.76	51.66 ± 4.70
	Assembly	81.21 ± 0.62	84.51 ± 1.87	81.81 ± 0.98	62.98 ± 3.74	55.84 ± 1.42	52.68 ± 1.89
VNet	Low res.	79.36 ± 2.14	82.24 ± 1.71	79.62 ± 1.22	60.53 ± 2.72	55.19 ± 2.85	52.56 ± 2.89
	Full res.	79.92 ± 1.05	82.73 ± 1.37	80.09 ± 0.95	64.46 ± 5.23	51.30 ± 3.56	50.14 ± 4.14
	Assembly	80.61 ± 0.37	84.10 ± 1.45	81.22 ± 0.64	64.62 ± 3.29	54.39 ± 1.26	52.99 ± 2.05

Metrics scores from five-fold cross validation.

dataset and $96 \times 160 \times 128$ pixels versus $64 \times 128 \times 128$ pixels for the Pancreas dataset.

3.6. Comparison to the State-of-the-Art

The proposed architecture was evaluated on three publicly available datasets: Task03-Liver, Task07-Pancreas from Medical Decathlon Challenge and the Kidney Tumor Segmentation 2019 Challenge dataset to compare its performance with state-of-the-art methods. Next three subsections summarize the results for each dataset.

3.6.1. Kidney

Our VNet with attention gates and deep supervision (VNet-AG-DSV) for the kidney-tumor task (**Table 7**) participated in the Kidney Tumor Segmentation Challenge of 2019, achieving a dice score 96.63 and 79.29 for kidney and tumor label, respectively, similar to our five-fold cross-validation values of 96.43 ± 1.06 and 79.94 ± 5.33 for kidney and renal tumor, respectively. The results show the stable transfer of values from validation to test set, which supports the stability of the model results. **Table 7** shows the test set results for three wining submissions compared to our model. The winning solution by Isensee and Maier-Hein (2019) uses residual 3DUNet. The major difference from our solution (apart from architectural model changes) is in the loss function, which was accommodated to fit the challenge scoring system. The authors also excluded some cases from the training set (this was allowed by organizers). Second (Hou et al., 2019) and third (Mu et al., 2019) submission in KiTS challenge use some variant of a multi-step solution, where the approximate position of the kidneys is determined in the first step and only then is produced the final precise segmentation map. Please note that we performed nor manual tweaking of the training set nor any accommodation to the challenge. We can then conclude that our VNet-AG-DSV showed remarkable performance with the

TABLE 5 | Comparison of the proposed VNet-AG-DSV to the state-of-the-art network with similar parameters presented by Isensee et al. (2018).

Model	MDC task03-liver		MDC task07-pancreas	
	Liver label	Tumor label	Liver label	Tumor label
Isensee et al. (2018)—Low res.	**94.69**	47.01	79.45	49.65
Isensee et al. (2018)—Full res.	94.11	**61.74**	77.69	42.69
Isensee et al. (2018)—Best model	95.43	61.82	79.30	52.12
VNet-AG-DSV—Low res.	94.54	**54.72**	**79.58**	**52.43**
VNet-AG-DSV—Full res.	**95.95**	57.65	**80.09**	**50.14**
VNet-AG-DSV—Best model	**95.74**	**64.70**	**81.22**	**52.99**

All the models were trained on the same dataset, released by Medical Decathlon Challenge (MDC) and validated in five-fold cross-validation. Higher score from the comparison of the two models is highlighted in bold.

TABLE 6 | Performance comparison.

	UNet	UNet-AG-DSV	VNet	VNet-AG-DSV
Num. parameters [M]	26.2453	26.2917	29.6873	29.7383
Train iteration* [ms]	224.8231	260.6527	297.2699	338.3336
Eval iteration* [ms]	189.7215	217.5776	268.6558	299.3836

**Measured as mean from 100 runs on GeForce GTX 1080 Ti.*

TABLE 7 | Test set results from the Kidney Tumor Challenge 2019 leaderboard.

Team	Composite dice	Kidney dice	Tumor dice
Isensee and Maier-Hein (2019)	91.23	97.37	85.09
Hou et al. (2019)	90.64	96.74	84.54
Mu et al. (2019)	90.25	97.29	83.21
VNet-AG-DSV	87.96	96.63	79.29

same architecture that was used for the other two previous tasks, namely detecting two other organs (pancreas and liver) along with their tumors (of a different structure to the kidney).

3.6.2. Liver

The liver-tumor dataset was obtained from the Medical Decathlon Challenge (MDC) happening at the MICCAI conference in 2018. We analyze the results from various research papers dealing with liver and liver-tumor segmentation. The Bilic et al. (2019) in work Liver Tumor Segmentation Benchmark (LiTS) presents a comparative study of two challenges dealing with liver and liver-tumor segmentation. Authors note that not a single algorithm performed best for liver and tumors simultaneously. The winner of liver segmentation, Tian et al. achieves the dice score 96.30 and 65.70 for liver and tumor class, respectively. The winner of the lesion segmentation part, Yuan et al. gained the dice score of 96.10 and 70.20 for the liver and tumor classes, respectively. All winning methods in LiTS benchmark utilized some post-processing steps, most commonly the connected component labeling but also other methods more specific for the concrete task of liver lesion detection. As shown in **Table 8**, our VNet-AG-DSV achieved the dice scores 96.37 and 64.70 for liver and tumor class, respectively. Our method, being fully automatic and not using hand-tuned post-processing, not only provides comparable results, it can also be easily transferred and used on different organ segmentations task as shown next.

TABLE 8 | Comprarison of the state-of-the-art methods for liver and liver-tumor segmentation from CT scans.

Team	Composite Dice	Liver Dice	Tumor Dice
Bilic et al. (2019)	83.15	96.10	70.20
Bilic et al. (2019)	81.00	96.30	65.70
Isensee et al. (2018)	78.63	95.43	61.82
VNet-AG-DSV	80.56	96.37	64.70

**The models were trained and tested on different dataset.*

TABLE 9 | Comprarison of the state-of-the-art methods for pancreas and pancreas-tumor segmentation from CT scans.

Team	Composite Dice	Liver Dice	Tumor Dice
Roth et al. (2018)*	-	81.27	-
Oktay et al. (2018)*	-	84.00	-
Isensee et al. (2018)	65.71	79.30	52.12
VNet-AG-DSV	67.11	81.22	52.99

**The models were trained and tested on different dataset.*

3.6.3. Pancreas

In comparison to other abdominal organs, the pancreas segmentation is a challenging task, as shown by the lower dice scores achieved in the literature. Roth et al. (2018) introduces an application of holistically-nested convolutional networks (HNNs) and achieves the dice score 81.27 ± 6.27. Oktay et al. (2018) introduces the attention gates for pancreas segmentation but compared to our solution does not include deep supervision while differing in other architectural choices. Their network achieves the dice score 84.00 ± 8.70 for the pancreas label. To best of our knowledge, there exist no papers dealing with both, pancreas and pancreas-tumor segmentation, except the ones submitted for the Medical Decathlon Challenge. The best dice score for the pancreas, and the pancreas-tumor segmentation, achieved in this challenge by Isensee et al. (2018) is 79.30 and 52.12, respectively. As shown in **Table 9**, the dice scores from our VNet-AG-DSV are 81.22 and 52.99 for pancreas and tumor label, respectively. Our method beats the nnUNet by Isensee et al. (2018) in both labels, and its pancreas segmentation result equals to the methods dedicated only to pancreas detection.

4. DISCUSSION

Conventional artificial neural networks with fully connected hidden layers take a very long time to be trained. Due to this, the convolutional neural network (CNN) was introduced. It is specifically designed to work with the images by the use of convolutional layers and pooling layers before ending with fully connected layers. Nowadays, convolutional neural network architectures are the primary choice for most of the computer vision tasks. CNN takes inspiration in biological processes in that the connectivity pattern between neurons corresponds to the organization of the animal visual cortex (Hubel and Wiesel, 1968; Fukushima, 1980; Rodríguez-Sánchez et al., 2015). Similarly, as in the eye, individual neurons respond to stimuli from a restricted (bounded by the filter size) region of the visual field. These restricted receptive fields of different neurons partially overlap, and together they cover the entire visual field.

Image segmentation is one of the most laborious tasks in computer vision since it requires the pixel-wise classification of the input image. Long et al. (2015) presents a cully convolutional neural network for image segmentation, firstly introducing the skips between layers to fuse coarse, semantic and local, appearance information. The work of Ronneberger et al. (2015) extended the idea of skip connections and applied it favorably in medical image segmentation. The possibility to examine the image at different image scales proved to be crucial in successful image segmentation. Due to a volume characteristic of medical data, the 3D variant of fully convolutional networks with skip connections was introduced by Milletari et al. (2016). This type of architecture is the most used CNN in the field of medical image segmentation since then, scoring best at most leading challenges dealing with the medical image segmentation in the last years: The Liver Tumor Segmentation Challenge in 2017 (Bilic et al., 2019), the Medical Decathlon Challenge in 2018 (Simpson et al., 2019), and the Kidney Tumor Segmentation Challenge in 2019 (Heller et al., 2019).

The deep supervision presented by Kayalibay et al. (2017) takes the idea of skip connections and uses it differently. It is a design where multiple segmentation maps are generated at different resolutions levels of the network. The feature maps from each network level are transposed by $1 \times 1 \times 1$ convolutions to create secondary segmentation maps. These secondary maps are not intended for any further refinement of the final segmentation map. Instead, it tries to correct the earlier layers of the network and "encourage" them to produce better segmentation results, thus speeding the convergence at training. The deep supervision is especially useful in tackling the problem of the vanishing gradient, which usually occurs during the training of very deep CNN.

Apart from the skip connections, many researches tried to incorporate the concept of attention into artificial CNN visual models (Mnih et al., 2014; Xiao et al., 2015; Xu et al., 2015; Chen et al., 2016). The presence of attention is one of the unique aspects of the human visual system (Corbetta and Shulman, 2002), which helps to selectively process the most relevant part of the incoming information for the task at hand. (Chen et al., 2016) proposes an attention model that softly weights the features from different input scales when predicting the semantic label of a pixel. Oktay et al. (2018) utilized a similar principle in their attention gates and applied them in medical image segmentation. Attention is especially helpful in the case of internal organ segmentation from abdominal computed tomography (CT) scans because abdominal organs are characteristically represented by similar intensity voxels in CT scans. The model greatly benefits from the ability to discard the activation from insignificant parts of the image and focus on the organ of interest. Eventually, the human expert would follow the same methodology: first, find the rough position of the organ of interest and only then analyze it in detail, as could be found in the description of the segmentation maps annotating process for the KiTS challenge (Heller et al., 2019).

5. CONCLUSIONS

This work presents a comprehensive study of medical image segmentation via a deep convolutional neural network. We propose a novel network architecture extended by attention gates and deep supervision (VNet-AG-DSV) which achieves results comparable to the state-of-the-art performance on several and very different medical image datasets. We performed extensive study which analyze the two most popular convolutional neural networks in medical images (UNet and VNet) across three different organ-tumor datasets and two training image resolutions. Further, to understand how the model represents the input image at the intermediate layers, the activation maps from attention gates and secondary segmentation maps from deep supervision layers are visualized. The visualizations show an excellent correlation between the activation present and the label of interest. The performance comparison shows that the proposed network extension introduces a slight computation burden, which is outweighed by considerable improvement in performance. Finally, our architecture is fully automatic and has shown its validity at detecting three different organs and tumors, i.e., more general than the state of the art, while providing similar performance to more dedicated methods.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: http://medicaldecathlon.com/, https://kits19.grand-challenge.org/.

AUTHOR CONTRIBUTIONS

AT and TT coded the proposed methodology and performed the experiments. ZK helped to ensure the needed computation power. AT wrote the first draft of the manuscript. AR-S did the first approval reading. All authors contributed conception and design of the study, contributed to manuscript revision, read, and approved the submitted version.

FUNDING

This work was supported by Internal Grant Agency of Tomas Bata University under the Project no. IGA/CebiaTech/2020/001, and by COST (European Cooperation in Science & Technology) under Action CA15140, Improving Applicability of Nature-Inspired Optimization by Joining Theory and

Practice (ImAppNIO). Finally, the access to computational resources supplied by the project e-Infrastruktura CZ (e-INFRA LM2018140) provided within the program Projects of Large Research, Development, and Innovations Infrastructures, is greatly appreciated.

ACKNOWLEDGMENTS

This paper was created with support of A.I. Lab (ailab.fai.utb.cz) from Tomas Bata University in Zlin and IIS group at the University of Innsbruck (iis.uibk.ac.at).

REFERENCES

Bilic, P., Christ, P. F., Vorontsov, E., Chlebus, G., Chen, H., Dou, Q., et al. (2019). The liver tumor segmentation benchmark (lits). *CoRR, abs/1901.04056*.

Chen, H., Dou, Q., Yu, L., Qin, J., and Heng, P.-A. (2018). VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images. *Neuroimage* 170, 446–455. doi: 10.1016/j.neuroimage.2017.04.041

Chen, L.-C., Yang, Y., Wang, J., Xu, W., and Yuille, A. L. (2016). "Attention to scale: scale-aware semantic image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 3640–3649. doi: 10.1109/CVPR.2016.396

Corbetta, M., and Shulman, G. L. (2002). Control of goal-directed and stimulus-driven attention in the brain. *Nat. Rev. Neurosci.* 3:201. doi: 10.1038/nrn755

Dou, Q., Chen, H., Jin, Y., Yu, L., Qin, J., and Heng, P.-A. (2016). "3D deeply supervised network for automatic liver segmentation from ct volumes," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Cham: Springer), 149–157. doi: 10.1007/978-3-319-46723-8_18

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* 36, 193–202. doi: 10.1007/BF00344251

Gibson, E., Giganti, F., Hu, Y., Bonmati, E., Bandula, S., Gurusamy, K., et al. (2018). Automatic multi-organ segmentation on abdominal ct with dense v-networks. *IEEE Trans. Med. Imaging* 37, 1822–1834. doi: 10.1109/TMI.2018.2806309

Grewal, M., Srivastava, M. M., Kumar, P., and Varadarajan, S. (2018). "Radnet: Radiologist level accuracy using deep learning for hemorrhage detection in CT scans," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)* (Washington, DC), 281–284. doi: 10.1109/ISBI.2018.8363574

Hamidian, S., Sahiner, B., Petrick, N., and Pezeshk, A. (2017). "3D convolutional neural network for automatic detection of lung nodules in chest CT," in *Medical Imaging 2017: Computer-Aided Diagnosis, Vol. 10134* (Orlando, FL: International Society for Optics and Photonics), 1013409. doi: 10.1117/12.2255795

He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Delving deep into rectifiers: surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision* (Santiago), 1026–1034. doi: 10.1109/ICCV.2015.123

Heller, N., Sathianathen, N., Kalapara, A., Walczak, E., Moore, K., Kaluzniak, H., et al. (2019). The kits19 challenge data: 300 kidney tumor cases with clinical context, CT semantic segmentations, and surgical outcomes. *arXiv [preprint]. arXiv:1904.00445*.

Hou, X., Xie, C., Li, F., and Nan, Y. (2019). *Cascaded Semantic Segmentation for Kidney and Tumor*. Technical report, PingAn Technology Co., Ltd, Shanghai. doi: 10.24926/548719.002

Hu, P., Wu, F., Peng, J., Bao, Y., Chen, F., and Kong, D. (2017). Automatic abdominal multi-organ segmentation using deep convolutional neural network and time-implicit level sets. *Int. J. Comput. Assist. Radiol. Surg.* 12, 399–411. doi: 10.1007/s11548-016-1501-5

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., and Liu, W. (2019). "CCNet: Criss-cross attention for semantic segmentation," in *The IEEE International Conference on Computer Vision (ICCV)* (Seoul). doi: 10.1109/ICCV.2019.00069

Hubel, D. H., and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *J. Physiol.* 195, 215–243. doi: 10.1113/jphysiol.1968.sp008455

Isensee, F., Kickingereder, P., Wick, W., Bendszus, M., and Maier-Hein, K. H. (2017). "Brain tumor segmentation and radiomics survival prediction: contribution to the brats 2017 challenge," in *International MICCAI Brainlesion Workshop* (Cham: Springer), 287–297. doi: 10.1007/978-3-319-75238-9_25

Isensee, F., and Maier-Hein, K. H. (2019). *An Attempt at Beating the 3D U-Net*. Technical report, German Cancer Research Center (DKFZ), Heidelberg. doi: 10.24926/548719.001

Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P. F., Kohl, S., et al. (2018). nnU-Net: Self-adapting framework for U-net-based medical image segmentation. *CoRR abs/1809.10486*. doi: 10.1007/978-3-658-25326-4_7

Jimenez-del-Toro, O., Müller, H., Krenn, M., Gruenberg, K., Taha, A. A., Winterstein, M., et al. (2016). Cloud-based evaluation of anatomical structure segmentation and landmark detection algorithms: visceral anatomy benchmarks. *IEEE Trans. Med. Imaging* 35, 2459–2475. doi: 10.1109/TMI.2016.2578680

Kayalibay, B., Jensen, G., and van der Smagt, P. (2017). CNN-based segmentation of medical imaging data. *CoRR abs/1701.03056*.

Kearney, V., Chan, J. W., Wang, T., Perry, A., Yom, S. S., and Solberg, T. D. (2019). Attention-enabled 3D boosted convolutional neural networks for semantic CT segmentation using deep supervision. *Phys. Med. Biol.* 64:135001. doi: 10.1088/1361-6560/ab2818

Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., and Tu, Z. (2015). "Deeply-supervised nets," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, Vol. 38 of Proceedings of Machine Learning Research*, eds G. Lebanon and S. V. N. Vishwanathan (San Diego, CA: PMLR), 562–570.

Linguraru, M. G., Richbourg, W. J., Liu, J., Watt, J. M., Pamulapati, V., Wang, S., et al. (2012). Tumor burden analysis on computed tomography by automated liver and tumor segmentation. *IEEE Trans. Med. Imaging* 31, 1965–1976. doi: 10.1109/TMI.2012.2211887

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 3431–3440. doi: 10.1109/CVPR.2015.7298965

Milletari, F., Navab, N., and Ahmadi, S.-A. (2016). "V-Net: fully convolutional neural networks for volumetric medical image segmentation," in *4th IEEE International Conference on 3D Vision (3DV)* (Stanford, CA: Stanford University), 565–571. doi: 10.1109/3DV.2016.79

Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. (2014). "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems* (Montreal, QC), 2204–2212.

Mu, G., Lin, Z., Han, M., Yao, G., and Gao1, Y. (2019). *Segmentation of Kidney Tumor by Multi-Resolution VB-nets*. Technical report, Shanghai United Imaging Intelligence Inc., Shanghai. doi: 10.24926/548719.003

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M. C. H., Heinrich, M. P., Misawa, K., et al. (2018). Attention U-net: Learning where to look for the pancreas. *CoRR abs/1804.03999*.

Rodríguez-Sánchez, A. J., Fallah, M., and Leonardis, A. (2015). Hierarchical object representations in the visual cortex and computer vision. *Front. Comput. Neurosci.* 9:142. doi: 10.3389/fncom.2015.00142

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Cham: Springer), 234–241. doi: 10.1007/978-3-319-24574-4_28

Roth, H. R., Lu, L., Lay, N., Harrison, A. P., Farag, A., Sohn, A., et al. (2018). Spatial aggregation of holistically-nested convolutional neural networks for automated pancreas localization and segmentation. *Med. Image Anal.* 45, 94–107. doi: 10.1016/j.media.2018.01.006

Sharma, N. K., and Aggarwal, L. M. (2010). Automated medical image segmentation techniques. *J. Med. Phys.* 35, 3–14. doi: 10.4103/0971-6203.58777

Simpson, A. L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., van Ginneken, B., et al. (2019). A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *CoRR, abs/1902.09063*.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., et al. (2017). "Residual attention network for image classification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI). doi: 10.1109/CVPR.2017.683

Winzeck, S., Hakim, A., McKinley, R., Pinto, J. A., Alves, V., Silva, C., et al. (2018). Isles 2016 and 2017-benchmarking ischemic stroke lesion outcome prediction based on multispectral MRI. *Front. Neurol.* 9:679. doi: 10.3389/fneur.2018.00679

Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y., and Zhang, Z. (2015). "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 842–850.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., et al. (2015). "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning* (Lille), 2048–2057.

Yang, G., Li, G., Pan, T., Kong, Y., Wu, J., Shu, H., et al. (2018). "Automatic segmentation of kidney and renal tumor in ct images based on 3d fully convolutional neural network with pyramid pooling module," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 3790–3795. doi: 10.1109/ICPR.2018.8545143

Ypsilantis, P.-P., and Montana, G. (2017). Learning what to look in chest x-rays with a recurrent visual attention model. *arXiv [preprint]. arXiv:1701.06452*.

Zhou, Y., Xie, L., Fishman, E. K., and Yuille, A. L. (2017). "Deep supervision for pancreatic cyst segmentation in abdominal CT scans," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Cham: Springer), 222–230. doi: 10.1007/978-3-319-66179-7_26

Zhu, Q., Du, B., Turkbey, B., Choyke, P. L., and Yan, P. (2017). "Deeply-supervised CNN for prostate segmentation," in *2017 International Joint Conference on Neural Networks (IJCNN)* (Anchorage, AK), 178–184. doi: 10.1109/IJCNN.2017.7965852

Zreik, M., Leiner, T., de Vos, B. D., van Hamersvelt, R. W., Viergever, M. A., and Išgum, I. (2016). "Automatic segmentation of the left ventricle in cardiac ct angiography using convolutional neural networks," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)* (Prague, CZ), 40–43. doi: 10.1109/ISBI.2016.7493206

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling Editor declared a past co-authorship with one of the authors AR-S.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 15 September 2020
doi: 10.3389/frobt.2020.00096

Improving Health Monitoring With Adaptive Data Movement in Fog Computing

Cinzia Cappiello [1], Giovanni Meroni [1], Barbara Pernici [1], Pierluigi Plebani [1], Mattia Salnitri [1], Monica Vitali [1], Diana Trojaniello [2], Ilio Catallo [2] and Alberto Sanna [2]*

[1] Dip. Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy, [2] Center for Advanced Technology for Health and Wellbeing, IRCCS San Raffaele Hospital, Milan, Italy

Pervasive sensing is increasing our ability to monitor the status of patients not only when they are hospitalized but also during home recovery. As a result, lots of data are collected and are available for multiple purposes. If operations can take advantage of timely and detailed data, the huge amount of data collected can also be useful for analytics. However, these data may be unusable for two reasons: data quality and performance problems. First, if the quality of the collected values is low, the processing activities could produce insignificant results. Second, if the system does not guarantee adequate performance, the results may not be delivered at the right time. The goal of this document is to propose a data utility model that considers the impact of the quality of the data sources (e.g., collected data, biographical data, and clinical history) on the expected results and allows for improvement of the performance through utility-driven data management in a Fog environment. Regarding data quality, our approach aims to consider it as a context-dependent problem: a given dataset can be considered useful for one application and inadequate for another application. For this reason, we suggest a context-dependent quality assessment considering dimensions such as accuracy, completeness, consistency, and timeliness, and we argue that different applications have different quality requirements to consider. The management of data in Fog computing also requires particular attention to quality of service requirements. For this reason, we include QoS aspects in the data utility model, such as availability, response time, and latency. Based on the proposed data utility model, we present an approach based on a goal model capable of identifying when one or more dimensions of quality of service or data quality are violated and of suggesting which is the best action to be taken to address this violation. The proposed approach is evaluated with a real and appropriately anonymized dataset, obtained as part of the experimental procedure of a research project in which a device with a set of sensors (inertial, temperature, humidity, and light sensors) is used to collect motion and environmental data associated with the daily physical activities of healthy young volunteers.

Keywords: data utility, fog computing, data movement, data analytics, data quality, quality of service

Edited by:
George Azzopardi,
University of Groningen, Netherlands

Reviewed by:
Eric Wade,
The University of Tennessee, Knoxville, United States
Alexander Lazovik,
University of Groningen, Netherlands

***Correspondence:**
Pierluigi Plebani
pierluigi.plebani@polimi.it

Specialty section:
This article was submitted to Sensor Fusion and Machine Perception, a section of the journal Frontiers in Robotics and AI

Received: *11 June 2019*
Accepted: *15 June 2020*
Published: *15 September 2020*

Citation:
Cappiello C, Meroni G, Pernici B, Plebani P, Salnitri M, Vitali M, Trojaniello D, Catallo I and Sanna A (2020) Improving Health Monitoring With Adaptive Data Movement in Fog Computing. Front. Robot. AI 7:96. doi: 10.3389/frobt.2020.00096

1. INTRODUCTION

The huge potential of the Internet of Things (IoT) paradigm has been immediately understood and applied in many domains (Ahmed et al., 2016) to provide advanced sensing layers, enabling solutions for personal needs (e.g., monitoring daily activities) (Alaa et al., 2017) as well as services for entire communities (e.g., smart cities) (Zanella et al., 2014). In the healthcare domain, IoT has also been adopted to similarly improve current processes and to provide new services to patients. However, extensive use of IoT data with continuous data flows from monitored patients can pose several challenges in developing effective and performing systems. For instance, the continuous and real-time monitoring of body parameters has so far been dedicated only to critically ill patients admitted to an intensive care unit. In any other case, to reduce the amount of transmitted data, either the patients are monitored at a lower rate (e.g., with a daily or weekly visit at the hospital) or, even if the sensors are able to constantly monitor patients, the monitoring data are collected from time to time (for example, the data are sent to the hospital every morning). In addition, the development of a system becomes more complex if we consider that the monitoring of a patient could require different types of sensors from different manufacturers, thus requiring a significant effort for integrating them and their underlying processes (Vitali and Pernici, 2016).

Although the IoT paradigm is addressing most of these challenges by considering the contribution of different communities, such as device manufacturers, network managers, internet-based solution providers, and semantic web researchers (Atzori et al., 2010), additional effort is required to create a more fruitful collaboration between the sensing layer and the application layer. While the former is focused on solving problems related to the observation and measurement of physical phenomena and the digital representation of such measurements, the latter is in charge of analyzing the sensed data to provide information and knowledge. Focusing on the deployment of this type of systems, the sensing layer is usually located on the edge, while the application layer is located on the cloud because it provides a more scalable and reliable infrastructure. On the other hand, there are situations in which the analysis (or a part of it) cannot be executed on the cloud. For example, data could not be moved from the premises for privacy reasons. Also, in the case of big datasets, moving all the data to the cloud for processing may take too long. Therefore, a more articulated deployment of the application layer—involving both the edge and the cloud—is required. In this context, Fog Computing (IEEE, 2018) has been introduced as a paradigm for creating applications able to exploit both the cloud and edge computational power as well as the devices in between to create a continuum between the two sides. This is particularly important in healthcare applications since data related to users are sensitive by definition. Their analysis and storage must therefore comply with current regulations, such as the General Data Protection Regulation (GDPR) (Ducato, 2016). At the same time, health monitoring solutions should be flexible with respect to the type of users. For instance, there are situations (e.g., emergencies) where the ability to provide rapid analysis is more important than having a 100% accurate result that could take an unacceptable period of time to be computed. Conversely, when the data are collected for diagnostic reasons, data accuracy is more important than their freshness.

The aim of this work is to present how the principles of the Fog Computing paradigm can be adopted to improve health monitoring with the aim of providing information at the right time, in the right place, and with the right quality and format for the user (D'Andria et al., 2015). For this reason, the proposed framework is based on two main elements:

- The *data utility* concept (Cappiello et al., 2017), which provides a quantitative evaluation of the relevance of the data obtained as the combination of two factors: (i) data quality, related to the fitness for use, which includes dimensions like accuracy, volume, and timeliness, and (ii) Quality of Service (QoS), related to the performance of the data delivery, which depends on the mutual location of where the data are stored and where they are used. Given a data source, not all the users have the same utility requirements of that data source. Also, the network can have different impacts on the user experience when accessing the data source. Given these assumptions, the data utility is assessed in two steps. First, a Potential Data Utility (PDU) is calculated to evaluate the data utility of a data source independently of a specific user. Second, when the user is known, the PDU is refined to obtain the actual data utility specific to the user's requirements.
- A *goal-based model* (Plebani et al., 2018), which is adopted to specify the requirements for the application layer with respect to the data utility. Compared with the typical goal-based models adopted in requirements engineering, the solution proposed in this paper also includes an additional *treatment layer*, following the methodology proposed in Vitali et al. (2015), which includes the adaptation actions available and the impact of the enactment of these actions on meeting the requirements.

Consequently, the combination of these two components offers to the Fog environment a tool to select the best strategy for copying or moving data between the different storage units whilst also considering the possible transformations required and the impact of the network. Given the requirements specified for the application, our framework reacts to their violations by selecting the best adaptation action. Since we are dealing with a dynamic environment, the best strategy as well as the impact of an action over the application requirements can change over time. The framework also takes this aspect into account.

The proposed approach was evaluated in the healthcare scenario, where the interests for data could vary for different users. For example, for a clinical expert monitoring a particular patient, the data must be detailed and promptly available. Conversely, when a clinician is performing data analysis for research purposes, coarse-grained data—requiring less network bandwidth—may be sufficient.

The rest of the paper is organized as follows. section 2 introduces the main characteristics of Fog Computing to the reader, while section 3 discusses the motivating example used throughout the paper. section 4 focuses on the data utility concept, and section 5 identifies the adaptation actions that could affect the data management. section 6 details the characteristics of the enriched goal-based model used to select the best adaptation actions, whose evaluation is illustrated in section 7. Finally, section 8 discusses related work on the data movement in Fog environments, and section 9 concludes the work outlining possible future work.

2. FOG COMPUTING

Fog computing is emerging as a paradigm for the design, development, implementation, and maintenance of applications that are not necessarily distributed in the same environment—either cloud or edge—but which could also be allocated to resources in between (e.g., cloudlets) (IEEE, 2018). This paradigm has been mainly conceived to have in mind IoT (Internet of Things)-based applications, which can be organized around four main layers: (i) the sensors and actuators, where the data are generated or actions to the environment have effects; (ii) the Monitoring and Control, where the state of the application is monitored and controlled; (iii) the Operational Support, where a deeper analysis of the produced data is performed; and (iv) the Business Support, where data about different environments are collected and analyzed as a whole. In particular, when referring to Fog computing, different deployment models can be adopted (see **Figure 1**) with the aim of exploiting not only the resources available on the cloud but also those on the edge and in the infrastructure layers connecting these two environments. Based on this configuration, when the scalability of a solution is a key issue, a cloud deployment is preferable, as it provides a virtually unlimited amount of resources. Conversely, when latency must be reduced as much as possible and/or privacy constraints require that data should not leave the location where they are generated, a deployment on the resources running on the edge is preferred.

On this basis, the Fog computing paradigm considers a data flow that mainly moves data from the edge, where the data are generated, to the cloud, where they are processed. However, devices on the edge are getting more and more powerful in terms of computational and storage resources. According to this, Fog computing takes advantage of these resources by distributing the computation among the layers. In each layer, data are processed and analyzed to provide a synthesis for the layer above. In this way, the amount of data that should be moved decreases layer by layer. Moreover, the resulting data aggregation enables a mitigation of the data privacy related issues.

Although there is consensus around this view of Fog computing, such a paradigm must be more than creating a data center in the box, i.e., Cloudlets (Satyanarayanan et al., 2009), to bring the cloud closer to data producers. Instead, Fog Computing must be seen as a "resource layer that fits between the edge devices and the cloud data centers, with features that may resemble either" (Varshney and Simmhan, 2017). In particular, as discussed in Bermbach et al. (2018), the principles of Service Oriented Computing can be valuable also for Fog-based solutions to create a set of services able to simplify the data management in a Fog infrastructure in terms of new abstraction models able to hide the details of smart devices living on the edge of the network that could be very heterogeneous. Moreover, the effort in Fog computing should also be focused on simplifying resource management while considering both edge and cloud resources.

In this direction, the DITAS project[1] is focusing on improving data-intensive applications by exploiting the peculiarities of Fog infrastructures, starting from the observation that most of the data, especially in IoT scenarios, are generated on the edge and are usually moved to the cloud to perform the required analyzes. While doing so can improve the performance of such data analysis due to the capacity and scalability of cloud-based technologies, there are situations in which this approach is not convenient or even impossible. For instance, when the amount of data to be analyzed is significant, the effect of the network may be considerable[2]. Furthermore, for privacy reasons, the owner of the data may not allow the movement of data outside of the boundaries of the organization unless they are anonymized and, in some cases, such an anonymization could hamper the analysis. On the other hand, limiting the computation to the resources on the edge could reduce the performance as the amount of resources, and their capacities are generally limited.

The depicted scenario is perfectly suited to the e-health domain where the data are heterogeneous (e.g., structured and unstructured data, images, and videos) and produced by heterogeneous devices; privacy is another a key issue, and the analysis of these data is complex, and, in some cases (e.g., during emergencies) it must performed quickly. Focusing on a single data analysis process, the adoption of the Fog computing paradigm can be helpful. Indeed, the computation can be organized hierarchically on the devices from the edge to the cloud, each of them specialized on some operations. Conversely, this approach cannot be so helpful in case there are many operators aiming to analyze in different ways the same dataset. In this case, there is a risk of having several deployments, each of them attempting to reach a local optimum, without any coordination in managing the common resources, like the computational power and the network bandwidth.

Focusing on the optimization of the data movement, it is fundamental to properly manage the information logistics (Michelberger et al., 2013), i.e., the delivery of information at the right time, in the right place, and with the right quality and format to the user (D'Andria et al., 2015). As a consequence, user requirements can be defined in terms of functional aspects, i.e., contents, and non-functional ones, i.e., time, location, representation, and quality (Plebani et al., 2017). To this aim, it is crucial to define a proper set of strategies to enable data management involving the resources in the Fog to enforce a given data utility (Cappiello et al., 2017). As shown in **Figure 2**, the DITAS project investigates the possibility to manage the deployment of applications which are based on the

[1] http://www.ditas-project.eu

[2] The influence of the network could be impactful to the point of making the data movement via network intractable. For this reason, Amazon offers a service called Snowball (https://aws.amazon.com/snowball/) to securely and efficiently move huge amount of data by physically moving the data storage devices.

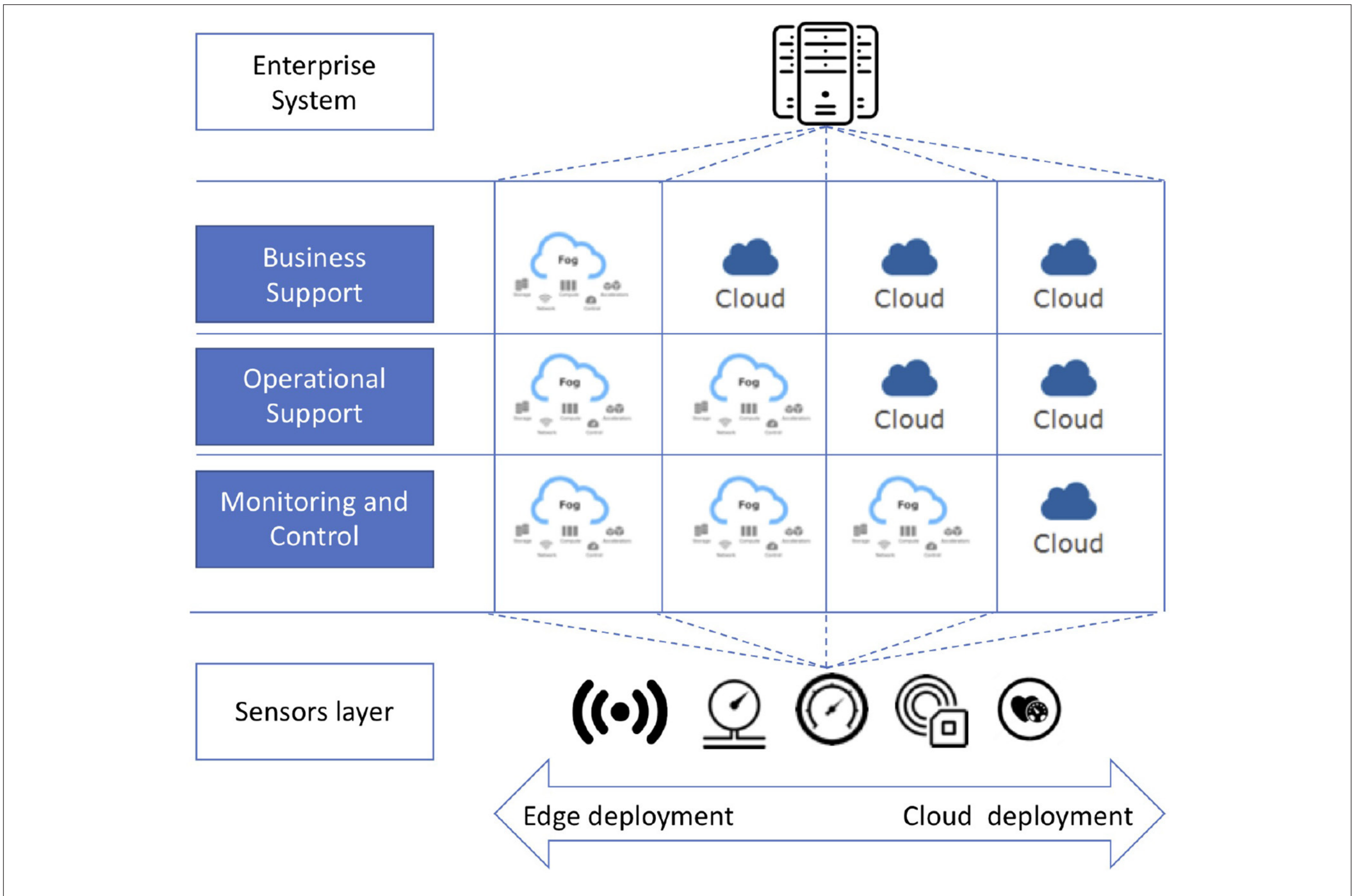

FIGURE 1 | Possible deployment models in Fog Computing (inspired by IEEE, 2018).

same data sources. In this way, the resources in the Fog for a given application can be organized according to a hierarchical topology. At the same time, when the same processing is required for different applications, the deployment approach could either go for a duplication of the computation nodes or allow a node to be shared among different applications. Regardless of the deployment strategy, which is out of the scope of this article, proper data management among the different nodes involved in all the considered applications is required due to the motivations discussed above. For instance, the data coming from the sensors can be collected in the data storage of a gateway close to the sensor layer. At the same time, different replicas of these data have to be put in place to serve some of the fog nodes. Since the type of computation performed on these nodes can vary, the frequency and type of data to be transmitted to these nodes can also vary.

3. MOTIVATING EXAMPLE

To properly introduce the approach proposed in this paper, we here describe a reference example, related to the usage of wearable devices as a means of facilitating patient monitoring. Indeed, in recent years, thanks to technology advances in the field of miniaturized sensors, various innovative wearable technologies have been developed. The introduction of such technologies in daily routines has raised great interest in new means of data collection in healthcare research and clinical contexts. Multiple applications for wearable devices have been identified in different areas of prevention, therapy, and well-being, ranging from the collection of relevant clinical data such as heart rate variability (HRV) to daily monitoring of physical activity. Furthermore, the possibility of collecting environmental parameters that could affect the subject's well-being through wearable devices is considered of great interest. In a recent H2020 project (I-SEE[3]), a new wearable device has been proposed that integrates a series of sensors, including UV, pressure, accelerometer, gyroscope, and light sensors. As depicted in **Figure 3**, in the scenario considered in this paper, the data collected by the wearable device are sent to a mobile application via Bluetooth. Part of the data processing is running on the wearable device and part on the mobile application. The data collected by the mobile application are daily (automatically) sent to the cloud. One of the main characteristics of the system is the presence of the UV sensor. Indeed, prolonged human exposure to solar UV radiation can have acute and chronic health effects on the skin, eyes, and immune system. In the long run, UV radiation

[3]https://isee-project.eu/

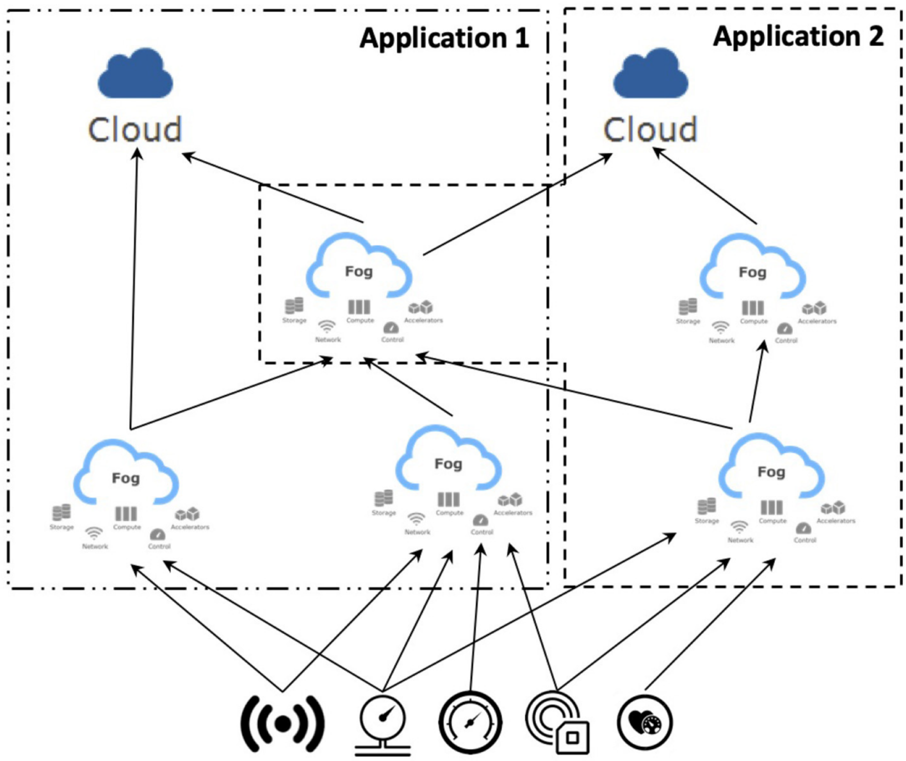

FIGURE 2 | Fog deployment in DITAS.

could also induce an inflammatory eye reaction. During outdoor activities, it is therefore important to be protected from UV rays to avoid their harmful effects especially for children, athletes, and individuals with the pre-maculopatia diagnosis. The availability of the UV sensor, correctly mounted on the wearable device, therefore allows the system to collect data on exposure to UV light during the day, consequently allowing for two interesting and clinically relevant applications:

- Self-monitoring: the UV sensor measures the amount of UV-A and UV-B and informs the user, through the mobile application, about the current UV exposure and related risk (on the basis of their risk profile, properly computed thanks to the user information collected through the mobile app). In case of overexposure, the mobile application alerts the user and proposes a solution in order to meet the compliance parameters. Since in the mobile application data are automatically saved in the cloud, the user could also verify, through a diary, their UV exposure condition over the last months. In addition, further useful insights for the user come from the combined analysis of the large amount of data collected by other users who experiment similar exposure conditions, e.g., the user can check their condition with respect to other people with similar risk profiles (e.g., age, sex, and photo-type) in the same geographical area. The user can also define a pool of other users (e.g., family members) who will be able to access their data.
- Expert-monitoring: the data collected and saved in the cloud could also be queried by clinical experts (e.g., dermatologists). Through a specific Web application, clinicians can remotely monitor patients and, when a risky condition occurs, invite them for a clinical visit. In this application, the possibility to access data of a large number of users allows the experts to obtain insights on different patients populations, based on the age, the sex, and other relevant features.

Both applications require the analysis of the collected raw data in order to extract meaningful information (i.e., UV intensity, time spent under UV, and time over risk thresholds). While the expert-monitoring application relies on the analysis of data collected over a long period (i.e., day, week, and month), thus not requiring a real-time analysis, the self-monitoring application aims to let the user be aware on their current condition/risk to get the proper information in real-time in order to act if alerted.

The self-monitoring application to achieve the objectives described is based on the characteristics of the users (i.e., age, gender, type of photo, etc.) and on the data collected by the sensors. These data are processed to (i) define the user profile (i.e., specific thresholds and risk factors) and ii) calculate the exposure. Note that the user profile is calculated on the mobile application since the user data are stored in the mobile device while the exposure calculation is implemented on the wearable device using the sensor data and then sent to the mobile application.

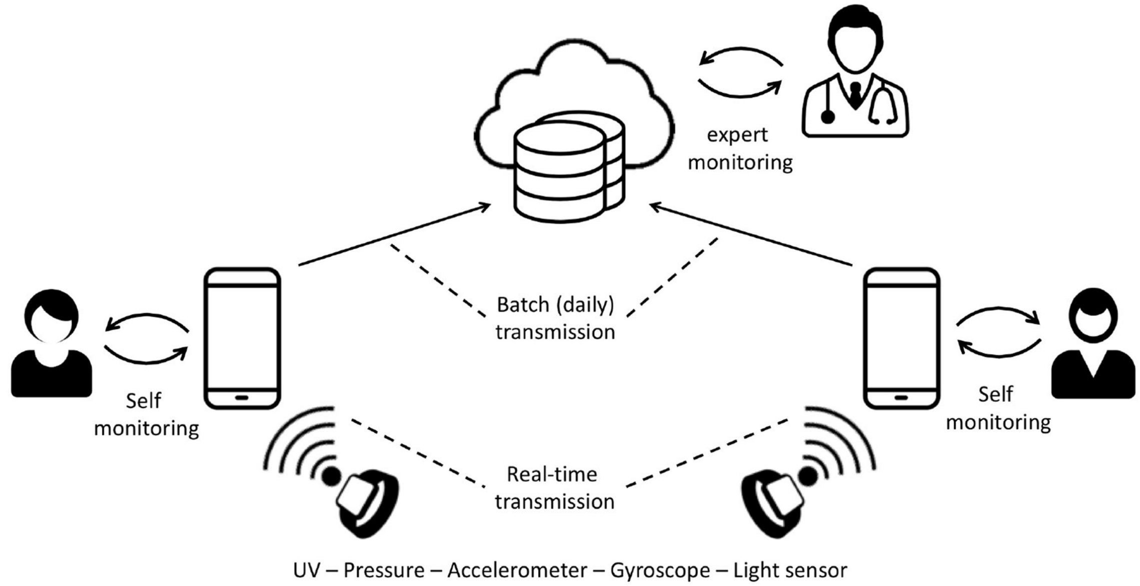

FIGURE 3 | Data flow in the motivating scenario.

4. DATA UTILITY

The two applications introduced in section 3 have different users and, in order to provide them with high-quality output at the right time and in the right place, these applications must select proper input data. For this reason, it is possible to associate each application with requirements relating to the data of interest and their granularity, as well as to the quality of the service (e.g., responsiveness to the request and availability). To represent these differences and support different applications in a customized way, we introduce the concept of *data utility*.

Data utility can be defined as the relevance of data for the usage context (Cappiello et al., 2017). Relevance is evaluated by considering the capability of the source to satisfy non-functional requirements (i.e., data quality and QoS properties) of the task using the data. Since data utility depends on the application/user that aims to access data for a specific goal, its assessment can be theoretically performed only when the usage context is defined. However, it is possible to identify and assess some dimensions in order to provide an objective estimation of the data source utility level, the so called *Potential Data Utility (PDU)*. The potential data utility provides an estimation of the quality level of the data contained in the whole data source. Note that the potential data utility and the data utility coincide when the user/application aims to use the entire data source as it is offered. As soon as the usage context is related to only a portion of the data source, the data utility must be assessed. However, the potential data utility can be seen as an aggregated reliability index of the data source. In order to assess the data utility, a set of relevant dimensions must be defined. In the following sections, data quality and QoS models are presented.

4.1. Data Quality Model

Data quality is often defined as "fitness for (intended) use" (Batini and Scannapieco, 2016), that is, the capability of a dataset to be suitable for the processes/applications in which it has to be used. Data quality is a multidimensional concept since different aspects of the analyzed data must be considered. Such aspects are modeled through data quality dimensions that are defined to analyze specific issues and that are assessed through determined metrics. The literature presents many data quality dimensions but, traditionally, the most used ones are

- *Accuracy*: the degree to which a value v is close to a correct value v' (Redman, 1996)
- *Completeness*: the degree to which all the values are present in the considered dataset
- *Consistency*: the degree of adherence to logical rules that link two or more attributes of the considered dataset
- *Timeliness*: the extent to which the age of data is suitable for the task at hand (Wang and Strong, 1996).

Note that the data quality model (i.e., the list of considered dimensions and the metrics for evaluating them) depends on the type of data source. For example, if we consider sensor networks, and therefore a scenario like the one considered in this paper in which the sources generate data streams, it is necessary to consider that the dataset DS is an infinite sequence of elements $DS = (X_1, t_1)(X_2, t_2)\ldots,(X_m, t_m)$ in which X_m is, for example, the set of values detected by the sensors on a wearable device at the moment t_m (Klein and Lehner, 2009). The model defined for data quality management relies on the concept of "data quality windows" for which data quality metadata are evaluated by dividing the stream in windows and assessing the quality of the k values included in a window. In this context, the metrics related to completeness, consistency, and timeliness do not change, while for the assessment of accuracy the maximum absolute systematic error a must be defined and a value v is correct if the expected value v' is in the range $[v - a; v + a]$ (Klein and Lehner, 2009). Accuracy is important, but it has to be considered together with *Precision* is the degree to which repeated measurements show the same or similar results. Precision is usually estimated by considering the standard deviation and might be an additional information to understand the stability of the measurement process. In fact, situations in which data are not accurate but precise may not always reveal malfunctioning sensors but also a plausible slow change in the observed phenomenon (for example, the expected temperature is increasing).

4.2. QoS Model

In the present work, the QoS model includes the following dimensions, which are the most commonly used in evaluating Quality of Service:

- *Availability*: it can be defined as "The ability of a functional unit to be in a state to perform a required function under given conditions at a given instant of time or over a given time interval, assuming that the required external resources are provided" (ISO/IEC, 2010). It usually shows the percentage of the time that the service is up and operational.
- *Response time*: it is the amount of time (usually expressed in seconds or milliseconds) the platform takes to provide the output of a specific request.
- *Throughput*: it is generally defined as the total amount of work completed in a given time. Considering data transmission, it refers to the data transfer rate.
- *Latency*: it is the time interval taken to transmit data between two points in a network.
- *Volume*: it is defined as the amount of disk space or the number of entries in a database.

The evaluation of all these dimensions requires a monitoring platform to provide this information as metadata associated with the data source.

4.3. Data Utility in Use

In our approach, each dataset DS is firstly associated with a *Potential Data Utility* vector:

$$PDU = (qd_1, qd_2 ... qd_N) \quad (1)$$

in which each value qd_i provides an estimation of a data quality or QoS dimension. As stated above, PDU is a set of metadata that profiles the source without considering the usage context. In this way, PDU provides aggregated information that helps users to understand the reliability of the dataset. PDU can thus be a first driver in the selection of sources if similar datasets are available.

As mentioned above, as soon as the context of use is related to a portion of the data source, it is necessary to evaluate the data utility. In fact, when a user aims to search for a dataset, they will define their functional and non-functional requirements. The former define the part of the available dataset that the user intends to access. Considering our scenario, in the self-monitoring application, the user could be interested only in the values collected in the last 10 min, while, for the expert monitoring application, the user can specify an interest for the data referred to a specific class of customers (for example, characterized by a specific profile, such as age or gender). Moving to the non-functional requirements, they refer to the constraints relating to a series of data quality/QoS dimensions (e.g., response time less than 5 s) considered relevant for the application/process in which the data are used. For example, accuracy, precision, completeness, and consistency are relevant dimensions for both applications, while timeliness is likely to be relevant only in the self-monitoring application where up-to-date data are needed. The description of the application together with the specified requirements define the usage context. In this second phase, the source can be associated with the *Data Utility* vector (DU) for a given usage context. DU informs users about the suitability of the dataset in satisfying their requirements. Note that PDU and DU overlap if the application/user asks to access the whole dataset, while DU has to be reassessed if the usage context considers a dataset $DS' \subset DS$.

At the run time, data utility should be periodically assessed in order to detect changes in the quality of data or service. In our approach, if the utility decreases below a certain threshold, one or more adaptation actions are triggered as described in section 5 with the goal of maintaining the data utility at a satisfactory level.

Note that, especially for what concerns the QoS criteria, data utility is dependent on the locations in which data are stored and consumed. In Fog computing, response time, for example, can significantly vary considering datasets in the cloud and datasets in the edge. By taking advantage of the ability to manage datasets in the edge and in the cloud and to move data between the different layers of the Fog computing environment, it is possible to trigger an adaptation action to continuously meet the requirements expressed. In section 5, such adaptation actions are formally defined and discussed.

5. ADAPTATION ACTIONS FOR DATA MANAGEMENT

A key feature of the proposed approach, in addition to the possibility to express the user requirements in terms of data utility, is to enforce the proper satisfaction of such requirements by enabling a set of adaptation actions that can be enacted to solve or prevent violations of the requirements. In particular, the actions considered in our approach refer to actions for moving or copying data, actions for improving the quality of data, and actions for transforming data to support or speed up data analysis.

Generally speaking, we refer to the actions that can be enacted to manipulate the data sources in a fog scenario as *adaptation actions*, which are composed of a set of atomic *tasks* $\mathcal{T} = \mathcal{DMT} \cup \mathcal{DTT}$, where (i) $\mathcal{DMT}$ are data movement tasks, and (ii) $\mathcal{DTT}$ are data transformation tasks (see **Figure 4**).

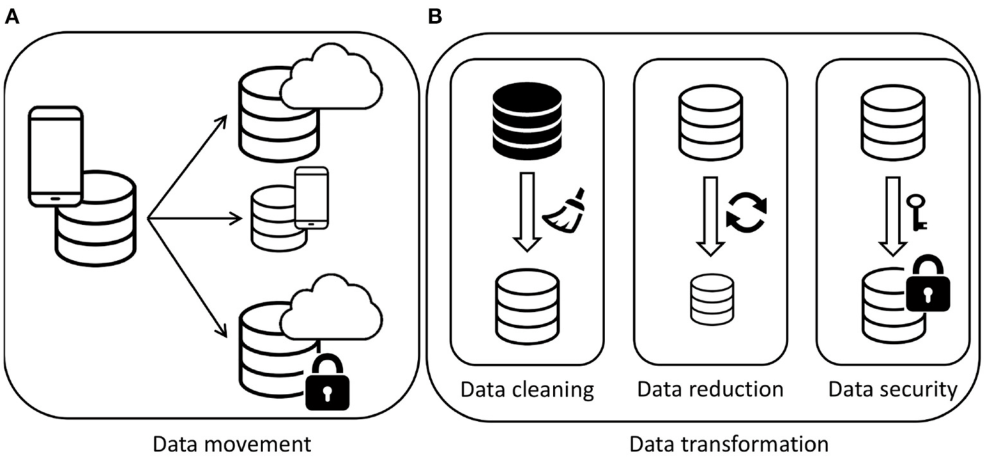

FIGURE 4 | Adaptation actions. **(A)** Data movement. **(B)** Data transformation.

In sections 5.1 and 5.2, we illustrate the atomic tasks for adaptations actions[4].

5.1. Data Movement Tasks

Data movement implies the transfer or the duplication of (a portion of) a dataset from a storage resource to another one. For instance, data can be moved from the edge—where they are generated—to cloud storage, like in the reference example where the data collected on a smartphone are periodically uploaded to the cloud. Generally speaking, a data movement task can be defined as

- The location of the resources involved in the movement from source a to destination b, which can be classified according to their layer (either in the edge E or in the cloud C)[5].
- The kind of movement applied, e.g., movement of data from one resource to another (deleting the previous version) M_{ab} or creation of a replica on different storage resources D_{ab}.

Considering the possible combinations of resource location and kind of movement, the following eight main relevant tasks are considered in $\mathcal{DMT}$:

- Move/Duplicate from cloud to edge (M_{CE}/D_{CE}): data are moved or copied from a cloud to an edge resource.
- Move/Duplicate from edge to cloud (M_{EC}/D_{EC}): data are moved or copied from an edge to a cloud resource.
- Move/Duplicate from cloud to cloud (M_{CC}/D_{CC}): data are moved or copied from a cloud to another cloud resource.
- Move/Duplicate from edge to edge (M_{EE}/D_{EE}): data are moved or copied from an edge to another edge resource.

Considering the running example, data movement between two edge devices occurs when data collected by one user is moved to another user's device, for example, to share information between family members. In addition, data movement between edge and cloud occurs when data about a user's activity is moved from their device to cloud storage. There, the data can be aggregated with the data of other users to be analyzed in the future by an expert.

The Data Movement tasks introduced in this section are categories of tasks. It means that they represent a generic movement according to the type of resources involved. In fact, categories are useful in aggregating together actions that are likely to have similar impacts when applied in a specific context. As discussed in Plebani et al. (2018), when implemented in a specific scenario and according to the actual resources available, one or more instances for each category might be instantiated to represent all possible movements between all the possible resources. Considering the example of **Figure 3**, two instances for the class M_{CE} and M_{EC} are created (since we have two edge devices connected with a cloud resource); similarly, two instances for M_{EE} are generated, and no tasks of type M_{CC} are available since only one cloud location is available in the scenario. The set of instantiated tasks depends also on the policies defined in the application context. For instance, in case we want to enable only movement in one direction, from edge to cloud, the M_{CE} tasks are not instantiated.

5.2. Data Transformation Tasks

While data movement tasks affect the location of a dataset, in the case of data transformation tasks a single data source is affected. In particular, the goal of this type of tasks is to produce a modified version of the dataset applying some filtering and/or transformations. More precisely, given a dataset DS where the degree (the number of domains) is $deg(DS)$, and the cardinality

[4]The following notation style is adopted: variables are strings in italic with a leading non-capital letter (e.g., x, y); sets are strings with the leading letter in the calligraphic font for mathematical expressions (e.g., $\mathcal{G}$, $\mathcal{I}$).

[5]For the sake of clarity and without losing generality, in this paper we assume in the following examples to have a Fog environment composed of only two layers, i.e., the edge and the cloud. When considering more layers, the formalization of the data movement slightly changes by introducing an index, which reflects the position on the hierarchy where the lower values represent the edge and the higher ones the cloud.

(the number of tuples) is $card(DS)$, a transformation task $dtt \in \mathcal{DTT}$ of a dataset DS produces a new dataset DS'

$$dtt(DS) \rightarrow DS' \tag{2}$$

On this basis, the transformation tasks could affect both the intensional and the extensional schema. In fact, in this class of tasks, data aggregation as well as data projections are included. In the first case, a set of tuples can be reduced to one (for example, by averaging a series of observations), which thus reduces cardinality. In the second case, some columns are removed as they are considered irrelevant or—in case of privacy problems—not accessible, thus reducing the degree of the dataset. Here we define three types of transformation tasks, which have different effects on the degree and the cardinality. In particular, we are interested in this work in three main sets of transformations: (i) *data-cleaning-related transformations*, (ii) *performance-related transformations*, and (iii) *security-related transformations*,

Data Cleaning related transformations aim to improve the quality of the data. In a data stream the cleaning tasks can be:

- **Inputting missing values**: missing values can be fixed by considering different techniques, such as (a) using unbiased estimators that estimate missing values without changing characteristics of an existing dataset (e.g., mean and variance), (b) using mean or median to replace missing values, or (c) adopting a specific distribution.
- **Outlier management**: an outlier can be generated since (a) the value has been incorrectly observed, recorded, or entered in a dataset, or (b) the value is correct but represents a rare event. The cleaning task is responsible for discovering outliers and for deciding between rare data and data glitches. Data glitches should be removed while rare events should be highlighted.

The former task mainly improves the completeness without negatively affecting the accuracy. In fact, the inputting techniques try to insert acceptable values. The latter task has instead a positive effect on accuracy and precision when data glitches are discovered. Note that a data cleaning task affects only the extensional schema of the data source, while the intensional schema is preserved. In fact, the improvement of data quality operates at record level [$deg(DS') = deg(DS)$ and $card(DS') = card(DS)$]. In summary, it is possible to enact a specific cleaning task on the basis of the dimension that caused a violation. The enactment of this task is time consuming and expensive in terms of computational power. Its execution performance might therefore be different in the edge or in the cloud.

In the considered scenario, data cleaning is a transformation technique that could be enabled both in the self- and expert monitoring when data quality requirements are not satisfied.

Performance-related transformations can be enacted to improve the performance of the enactment of an adaptation action. One of the main issues with adaptation actions is the management of high volumes of data that can generate delays and performance issues. For instance, the volume of data collected by the IoT and sensors at the edge makes the data movement for analysis from the edge to the cloud difficult and time consuming, and, in addition, it might introduce critical delays. For this reason, it is important to reduce the size of the data to be moved in order to make this task more agile. Performance related transformations are:

- **Aggregation**: the content of a data storage is reduced using aggregation operations (e.g., average, maximum, and minimum) summarizing several tuples.
- **Reduction**: the data volume is reduced by exploiting relations among data.

Both performance transformations reduce the volume of the data that must be transferred from the source to the destination.

Aggregation applies classical operators (e.g., average, maximum, minimum, and sum) to several events collected in the dataset. The effect is to reduce the volume of the dataset while affecting the level of detail contained in it. Aggregation is not reversible (it is not possible to obtain the original data from the aggregated set). This transformation aims to reduce the cardinality of the dataset but it does not affect the degree ($deg(DS') = deg(DS)$ and $card(DS') < card(DS)$).

Reduction is based on the assumption that the information stored in a dataset may contain related items. In literature, data reduction is performed mainly on a single signal by varying the sampling frequency based on the variability of the monitored variable (Trihinas et al., 2018). We additionally propose to exploit relations between different variables, which can be expressed as dependencies among the values of related attributes. For example, relations are those expressed through functional dependencies between the data values in a dataset. Functional dependencies are also used to check consistency in the dataset. For instance, it is possible to obtain the city and the country where a user resides from the postal code. A causal relation between attributes in a dataset can be expressed through an association rule $A \implies B$ expressing that the value of attribute B depends on the value of attribute A. Association rules are effective to represent relations between non-numerical attributes that can get a limited number of values. For numeric values, we instead apply regression functions to represent the dependencies between a dependent variable and a set of correlated variables from which it is possible to calculate its value. Relations can be both explicit and implicit. Explicit relations are declared by the data owner, who also provides the association rules for non-numerical attributes or the regression model for numerical values. Instead, implicit relations can be detected using data mining and machine learning techniques. While several approaches exist for extracting association rules between attributes of a dataset, the detection of dependencies between numerical values is not trivial (Peng and Pernici, 2016). Reduction enables to regenerate the original information, although with some approximations. It might reduce both the degree and the cardinality ($deg(DS') \leq deg(DS)$ and $card(DS') \leq card(DS)$).

Security-related transformations aim to satisfy security constraints that might affect a data source when moved from one location to another. As an example, when data are collected inside the user device, they contain the information that is needed to identify a specific person. When these data need to be moved and stored outside the device, privacy constraints might require that sensitive information must be hidden so that unauthorized customers cannot access it. Security-related transformations include the following.

- **Pseudonymization**: data are manipulated to substitute identifying fields within a data record with artificial identifiers.
- **Anonymization**: data are manipulated to remove all possible identifiers.
- **Encryption**: the data contained in a data storage are manipulated using encryption algorithms to make them unreadable to unauthorized users.

None of the security-related transformations affect the cardinality of the dataset ($card(DS') = card(DS)$). Instead, while pseudonymization and encryption do not affect the degree, anonymization might remove some of the attributes from the dataset ($deg(DS') \leq deg(DS)$).

5.3. Defining Relevant Adaptation Actions

Based on the knowledge of the available datasets, their location, their relations, and the privacy and security constraints, there are two main aspects that the system design has to take into account to define suitable adaptation actions based on the atomic tasks illustrated above:

- Defining the adaptation actions that are relevant in the considered application domain
- Identifying the most suitable action to perform in a given time.

In this section, we focus on the first problem, discussing the relevant aspects that must be taken into consideration when adaptation actions are to be finalized. About the second issue, section 6 discusses how the goal-based approach is adopted to drive the execution of the adaptation actions with the aim to improve data utility.

The different kinds of tasks introduced in this section are the building blocks for composing an adaptation action. In fact, an adaptation action can include one or more tasks. More formally, an adaptation action $aa \in \mathcal{AA}$ in a specific application context is defined as a tuple:

$$aa =< t_a, ManT, OptT > \tag{3}$$

where

- $t_a \in T$ is the main task of the action and can be either a data movement or a transformation task.
- $ManT \in \mathcal{DTT}$ is a set of mandatory tasks that are always executed with the main task.
- $OptT \in \mathcal{DTT}$ is a set of optional tasks that can be associated with the main task.

Both mandatory and optional tasks are transformations applied to the dataset for complying with the security requirements or for improving the effect of the main task. As an example, different privacy and security constraints might apply to each location, and a data movement action could therefore also require some security-related data transformation. For instance, due to the privacy regulations, data stored on the cloud must be anonymized, and data collected on a smartphone should thus be made anonymous by removing any direct reference to the user (e.g., userid and name) before moving them from the user's device to the cloud. According to this, we can define an adaptation action $aa_1 = \{M_{EC}, \{anonymization\}, \{reduction, aggregation\}\}$ composed of a main task $t_a = M_{EC}$, which moves the data collected by a wearable device from the smartphone of the user to the cloud storage. The mandatory task *anonymization* forces to anonymize the data when the movement is performed. Finally, $Opt = \{reduction, aggregation\}$ defines as optional the tasks reduction and aggregation, both reducing the volume of data to be moved from the device to the cloud, and this consequently improves latency and reduces cost.

As already defined, adaptation actions might affect both the content (data transformation tasks) or the location (data movement tasks) of a dataset. The argument of an adaptation action can be a whole dataset or a subset of it. As an example, when the cloud is fed with the data from the user's device, the action could include either a M_{EC} or a D_{EC}. Considering our scenario, when the storage on the device is almost full, moving data from the edge to the cloud might require emptying all the collected data stored in the edge to be saved in a cloud resource. However, the requirements of the running applications might be in conflict with this strategy since some data might be useful locally. For instance, some of the data should be kept locally to support the self-monitoring application. Observing the past executions of the application it is possible to provide information on the typical behavior expected by the system. Here, we focus on two main aspects:

- Relevant data: not all the data collected by the sensors at the edge are used locally. When deciding which data to move from the edge to the cloud and vice versa, we should take into account the frequently accessed data. This information is relevant to improve the performance of the data retrieval. As an example, when the storage resource at the edge side is full, we should move some data to the cloud. In doing so, we can select the data that are less likely to be used in the near future and keep the other data on the device to keep the data retrieval latency low.
- Device behavior: in a fog environment, we are often subject to unreliable connections between the cloud and the edge. Consequently, the user device can be offline at some point, making the communication between the cloud and the edge impossible. Observing the typical behavior of the devices in terms of connectivity with the cloud, we can prevent connectivity issues by using this information when deciding where to place the data. As an example, statistics and aggregation of monitoring data are usually performed and stored in the cloud. A customer who wants to access statistics needs to have an active connection all the time. If a permanent active connection is not ensured, our approach can improve the performance by saving an extract of the statistics back to the edge, thus making it accessible every time to the customers, even when connectivity is not present.

6. IMPROVING E-HEALTH MONITORING WITH DATA-UTILITY DRIVEN ADAPTATION ACTIONS

The main goal of each data provider is to offer its services by meeting consumer demands in terms of data usage. However, Fog computing is a dynamic environment in which the performance of the fog nodes can deteriorate and the connections between the nodes are not reliable or simply not durable due to the mobility of the fog nodes. This section describes the part of the approach, proposed in this document, which allows the provider to meet the users' data requirements in such a dynamic environment, choosing the best adaptation action.

We define a goal-based modeling language in order to specify (i) user requirements, (ii) the adaptation actions that can be implemented, and (iii) the link between adaptation actions and user requirements. The information modeled with this language is the basis for selecting the best adaptation action.

In the literature, several goal-based modeling languages are defined (Horkoff et al., 2015). However, as far as we know, no goal-based modeling language allows the definition of adaptation actions and their impact on goal models. We based the definition of the language on our extension (Plebani et al., 2018) of BIM modeling language (Horkoff et al., 2012).

6.1. A Modeling Language to Link User's Requirements With Adaptation Actions

Non-functional requirements of applications using the same dataset are expressed through the concept of data utility introduced in section 4. For each application, a different set of dimensions is selected, and the desired value is indicated for each dimension. When these requirements are not met, our approach identifies the violations and detects which adaptation actions can be enacted.

We have chosen a goal-based modeling language to represent the requirements since this type of language allows for easy classification of requirements based on users' objectives (goals) across different levels of abstraction. This feature allows the readability of the goal model even by non-technical users.

In a goal model, the *goal* concept represents an objective to be achieved. Formally, the set of goals $\mathcal{G}$ in a goal model is defined as

$$\mathcal{G} = \{< Name, \mathcal{M}etrics >\} \quad (4)$$

where *Name* is the name of the goal and $\mathcal{M}etrics$ a set of metrics used to assess the goal defined as

$$Metric = \{< Type, Comparator, Measure >\} \quad (5)$$

where $Type \in Types$ indicates the type of the metric referring to the set of data utility dimensions defined in section 4; $Measure \in \mathbb{R}$ represents the reference value; $Comparator \in \{<, \leq, >, \geq, =\}$ represents the relation between the observed value and the reference value.

In a goal model, each goal can be decomposed into sub-goals forming a tree structure, where the root element is called root goal. Sub-goals represent a set of objectives that, once achieved, allow the achievement of their parent goal. Root goals specify the main objectives (requirements) of users and, therefore, must be satisfied.

The upper part of **Figure 5** provides a graphical representation of the goal-based model formalization applied to the self-monitoring application described in section 3. Each ellipse represents a goal that is linked to measurable metrics that are used to specify when a goal is achieved. For example, the "High availability" goal is linked to the "Availability >99.5%" metric, which means that the goal is considered achieved if the availability of the service is more than 99.5%. The diagram shows two goal trees. On the left, the goal tree is composed only of a "Light client" goal, which specifies that the user wants to limit the volume of the data stored locally in the edge device. On the right, the target tree represents the user's data utility requirements, and it is more complex since it contains a decomposition of the goal model. For example, "Quality of Service" is a parent goal, while "Fast response" and "High accessibility" are its sub-goals. Goal models define two types of decomposition:

- AND-decomposition: all sub-goals must be achieved to achieve the parent goal;
- OR-decomposition: at least one sub-goal must be achieved to achieve the parent goal.

It is worth noting that a violation of a single metric—and therefore of the linked goal—may not imply the violation of the root goal. For example, if the latency of the provided service goes above 50 ms while its availability is greater than 99.5%, then the user's requirements are not violated since the two metrics are linked to two goals which OR-decompose the parent goal. The goal model in the figure requires only one of them to be achieved.

The set of all decompositions are represented by the set $\mathcal{D}ecompositions$ that is defined as

$$\mathcal{D}ecompositions = \{< g, sub, type >\} \quad (6)$$

where:

- $g \in Goals$, is the parent goal;
- $sub \in \mathcal{P}(Goals)$, belongs to the power set (i.e., the set of all possible combinations of the elements) of goals and contains the children goals that decompose the parent goal;
- $type \in \{and, or\}$ is the type of decomposition.

For example, the goal "High data utility" is AND-decomposed in three goals: "High quality of service," "Privacy," and "High data quality." This results in the following decomposition:

$$\{High\ data\ utility, \{High\ quality\ of\ service, Privacy, High\ data\ quality\}, AND\}$$

The lower part of the model in **Figure 5** represents the adaptation actions that can be enacted in this running example. The adaptation actions modeled define movement and duplication between edge devices to cloud storage, and data transformations. Adaptation actions are represented by boxes with a label that defines the source, the destination, and the type of action. All

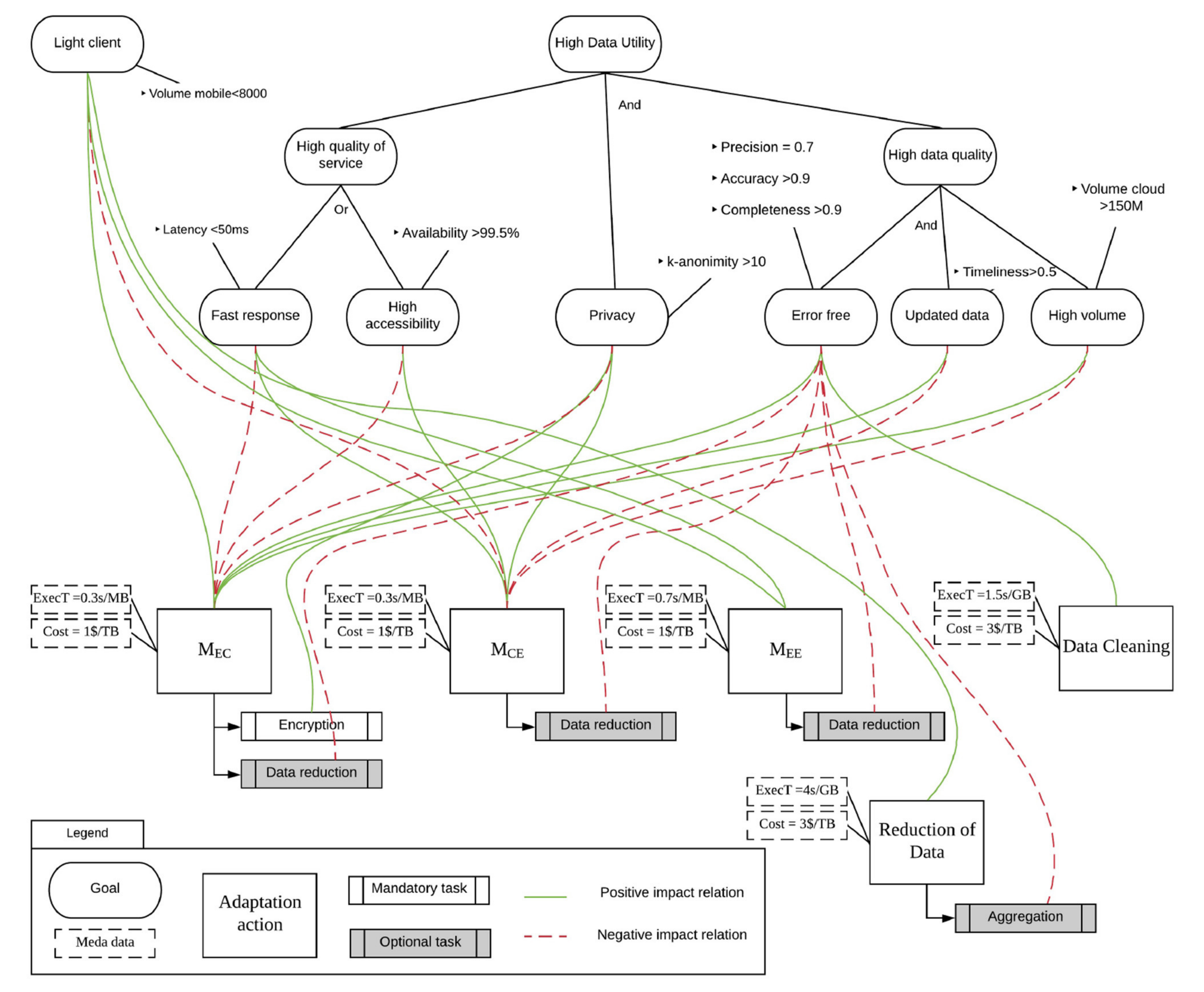

FIGURE 5 | Example of a goal model.

tasks composing the adaptation action are linked to goals with relations that specify the positive or negative impact that their enactment has on the achievement of the goal. For example, the adaptation action M_{EC}, according to the definition given in section 5.3, defines an action in which the main task concerns the movement between the sensors and the cloud. The action is linked with a positive impact on the "High volume" goal since the movement of data in the cloud has a positive impact on the linked metrics. Indeed, storing data in the cloud, instead of in an edge device, enables the storage of a higher volume of data.

As specified in section 5.3, an adaptation action is associated with optional and mandatory tasks. In the modeling language proposed in this paper, these tasks are represented with a box with double borders on both sides associated with an action: mandatory tasks have a white background, while optional tasks have a gray background.

Each action is associated with a link to the goals that represents the impact of the action over the goal satisfaction. Impacts can be positive or negative. For each action, $\mathcal{P}os \subseteq Goals$ represents the set of goals that receive a positive impact when the main task is executed, and $\mathcal{N}eg \subseteq Goals$ represents a set of goals negatively impacted. Also, $\mathcal{P}os \cap \mathcal{N}eg = \emptyset$. Optional and mandatory tasks inherit impacts of the adaptation actions they are linked to. If they provide additional or different impacts, links to the affected goals are represented in the model. For example, the adaptation action M_{EC} has one mandatory task "Encryption," which specifies that data can be encrypted before moving them. Such task has a positive impact on the "Privacy" goal since the encryption will prevent the disclosure of personal data. In this example, the adaptation action and the linked task have an opposite impact on the "Privacy" goal. If this is the case, the impact of the task overcomes the impact of the main task of the adaptation action.

Impact relations can be designed by experts or learned/refined automatically by observing the effects of executing a task on the metrics linked to goal model.

As specified in section 5.3, adaptation actions are enriched with metadata. These metadata in our model specify two aspects: the execution time of the action and the economic cost of its execution. Metadata are represented in the model as dashed boxes. For example, the adaptation action M_{EC} has attached two metadata specifying the execution time (i.e., an estimation of 0.3 s per MB) and the cost (i.e., 1$ per TB). According to this, for each action we can define an element of the metadata set $metaData \in \mathcal{M}etaData$ as

$$metaData =< metaType, \mathbb{R} > \tag{7}$$

where $metaType \in \{ExecT, Cost\}$ is the set of possible metadata. Optional and mandatory tasks can change the metadata of the action to which they are linked.

Formally, we define a task $t \in \mathcal{T}$, including impact relations and meta data:

$$t =< taskType, \mathcal{P}os, \mathcal{N}eg, \mathcal{M}etaData > \tag{8}$$

where $taskType$ is the type of task, as defined in section 5, $\mathcal{P}os$ and $\mathcal{N}eg$ are the positive and negative impacts of the task, and $\mathcal{M}etaData$ are the metadata associated with the task.

At this point we can define a goal model $\mathcal{GM}$:

$$\mathcal{GM} =< \mathcal{G}, \mathcal{D}ecompositions, \mathcal{AA} > \tag{9}$$

6.2. Creation of A Goal-Based Diagram for Supporting Adaptation Action Selection

The creation of a diagram based on the modeling language defined in section 6.1 consists of the following phases:

- Creation of the goal model structure that represents user requirements;
- Specification of the adaptation actions that can be enacted, complemented with mandatory and optional tasks and metadata;
- Specification of the impact of adaptation actions and tasks on the goal model.

In terms of the specification of the goal model, users specify their requirements based on their objectives. As already described before, we chose a goal-based modeling language since it can be used to represent requirements at different levels of abstraction. Users can thus express abstract requirements in the upper part of the goal model, while they can specify more concrete requirements on the lower part nearer to the leaves and up to the definition of the reference values for the metrics.

The specification of the adaptation actions largely depends on the infrastructure and on the resources. In section 5.3, we defined a set of classes that will be instantiated according to the actual context of execution. Instances of adaptation actions are not specified in **Figure 5** due to space constraints. For example, the adaptation actions M_{EE}, which consists in the movement between two edge devices, will be instantiated by generating two actions for each possible pair of edge elements authorized to exchange the data.

6.3. Automated Selection of Adaptation Actions

The main objective of the proposed goal-based model is to provide a method for identifying which is the best action to be enacted in case a goal is violated. In fact, when the violation of a metric prevents the achievement of the top goal of a goal tree, the model supports the selection of the best adaptation action to be implemented in order to remove the violation. This is reflected by the connections between the upper and the lower layers of the model. A software component has been developed[6] supports this identification by exploring the tree and the positive/negative impacts. The selection is divided into the three phases described below. The software requires the implementation of a monitoring system that identifies the violations of user requirements.

Phase 1: Selection of the relevant adaptation actions The first step is to identify the set of adaptation actions that can be enacted to solve a violation. In this step, impacts are used to identify the actions with a positive effect on the violated goal. For example, if the latency goes above 50 ms, the requirement specified by **Figure 5** for the sub-goal "Fast Response" is violated. Two adaptation actions are then selected: M_{EE} and M_{CE}.

This phase executes two algorithms: the first one identifies the violated goals (Algorithm 1), while the second one identifies the adaptation actions that have a positive impact on them (Algorithm 2).

Algorithm 1 requires as input the goal model $\mathcal{GM} = \{\mathcal{G}, \mathcal{D}ecompositions, \mathcal{AA}\}$ defined in Equation (9), and a set of measures $\mathcal{M}e = \{T_{me}, M_{me}\}$, where $T_{me} \in Types$ and $M_{me} \in \mathbb{R}$. Measures are generated by a monitoring system that continuously checks the system targeted by the goal model.

Algorithm 1: Identify unsatisfied goals

1: $\forall\ (g, \mathcal{M}etrics) \in \mathcal{G}$
2: $\quad \forall\ (t_m, C_m, M_m) \in \mathcal{M}etrics$
3: $\quad\quad$ if $\exists\ m_e = (t_{me}, m_{me}) \in \mathcal{M}e$ s.t. $t_{me} = t_m \wedge \neg apply(m_{me}, c_m, m_m)$
4: $\quad\quad$ then g is violated

where $apply(m_1, c, m_2) : boolean$ is a function that applies the operator $c \in \mathcal{C}omparators$ to the two measures $m_{me}, m_m \in \mathbb{R}$.

Algorithm 1 inspects all goals in the goal model (Line 1). For each metric in each goal (Line 2), the algorithm checks if it exists a measure that has the same type and violates one of its metrics (line 3). If this is the case, the goal is considered to be violated (Line 4).

Given a goal model $\mathcal{GM} = \{\mathcal{G}, \mathcal{D}ecompositions, \mathcal{AA}\}$ and the set of violated goals $\mathcal{VG}$ identified thanks to Algorithm 1, Algorithm 2 identifies the relevant adaptation actions in $\mathcal{AA}$.

In Algorithm 2:

[6]https://github.com/DITAS-Project/decision-system-for-data-and-computation-movement/

Algorithm 2: Identify adaptation actions

1: $\forall(t_a, ManT, OptT) \in \mathcal{AA}$
2: if $\exists g \in \mathcal{VG}$ s.t.$g \in (pos(t_a) \cup poss(ManT)) \setminus negs(ManT)$
3: then $(t_a, ManT, \emptyset) \rightarrow \mathcal{S}electedAA$
4: $\forall o_p \in OptT$
5: if $\exists g \in \mathcal{VG}$ s.t.$g \in pos(o_p)$
6: then $(t_a, ManT, \{o_p\}) \rightarrow \mathcal{S}electedAA$
7: $\forall selectedAA = (t_a, ManT, OptT) \in \mathcal{S}electedAA$
8: if $\exists g \in \mathcal{VG}$ s.t. $g \in$
9: $(neg(t_a) \setminus (poss(ManT) \cup poss(OptT)))\cup$
10: $(negs(ManT) \setminus poss(OptT))\cup$
11: $negs(OptT)$
12: then remove *selectedAA* from $\mathcal{S}electedAA$

- $pos(T) : Goals$, where $T \in \mathcal{T}$ is a function that returns the set of goals positively impacted by the task in input;
- $neg(T) : Goals$, where $T \in \mathcal{T}$ is a function that returns the set of goals negatively impacted by the task in input;
- $poss(\mathcal{T}') : Goals$, where $\mathcal{T}' \subseteq \mathcal{T}$ returns the union of all goals positively impacted by all tasks in the set received as input;
- $negs(\mathcal{T}') : Goals$, where $\mathcal{T}' \subseteq \mathcal{T}$ returns the union of all goals negatively impacted by all tasks in the set received as input.

Algorithm 2 considers each adaptation action (Line 1) and verifies if there exists a violated goal that belongs to the set of goals that are impacted positively by the main task or the mandatory tasks of the adaptation action (Line 2). If this is the case, the adaptation action is added to the set of selected actions *selectedAA*. The selected action set includes only the main task and the mandatory tasks (Line 3). Please notice that, in Line 2, we subtract the goal negatively impacted by *ManT* tasks since impacts of mandatory and optional tasks override impacts of the main task (see section 6.1).

Additionally, for each optional task of the action (Line 4) the algorithm verifies if they have a positive impact on a violated goal in $\mathcal{VG}$ (Line 5). If this is the case, the adaptation action is selected, including the optional task.

Finally, for each action in *selectedAA* (Line 7), the algorithm checks if the action has a negative impact on one of the violated goals (Line 8–11), removing it from the set (12). Similarly to Line 2, also in this case the algorithm subtracts from the set of goals negatively impacted by the main task the positive goals of mandatory and optional task (Line 9). Line 10 specifies that an impact relation of an optional goal overrides an impact relation of a mandatory goal. We chose this criterion since we assume optional tasks are chosen to improve the behavior of the adaptation action (and its mandatory tasks).

Phase 2: Prioritization of the adaptation actions The second phase consists of the prioritization of the adaptation actions selected in the first phase based on their metadata and on the strategy selected by the user. Strategies are optimization functions that consist of the (set of) metadata that the user would like to minimize or maximize. For example, they can define as a strategy the minimization of the costs or the minimization of the execution time. Once the strategy has been defined, the selected adaptation actions will be ranked and the enactment of the action with the highest score suggested.

The selection, and consequently the enactment, of an adaptation action brings to the system a new configuration where data have been moved, copied, and/or transformed to resolve a violation. Algorithm 2 selects the actions that have a positive impact on violated goals. It is worth noting that the correctness of the output, i.e., whether the selected adaptation action positively impacts the violated goals as expected, is based on the correctness of the input, i.e., the goal model analyzed.

Phase 3: Update of impact relations After the enactment of an adaptation action, the framework will periodically check the metrics and update the impact relations based on the performance of the system after the enactment.

□

Adaptation actions are selected and enacted one at time, with a time span between two enactments that is sufficient to measure the impact of the action on the goal model. Every time an action is selected by Algorithm 2, it is enacted, and metrics of the goal model are measured in order to detect the impacts of the adaptation action and, possibly, update its impact relations.

Algorithm 2 selects only adaptation actions with a positive impact on the goals violated in the model. This ensures that, granted the correctness of the goal model, the system is led to a configuration that resolves (or reduces) the violations. This is guaranteed by lines 1–3 of the Algorithm 2 where only actions with positive impacts on violated goals are selected. Lines 4–6 enrich this set of actions with actions with optional tasks, having at least a positive impact on violated goals, while lines 8–11 remove from the set adaptation actions with negative impacts on the violated goals. This last step aims to avoid side effects in the enactment of an action.

It is worth noticing that the method described in this section is successful only if the goal model is generated in a proper way. First of all, the goal model must contain all the relevant requirements of the application/user. It is very important to be able to capture whether the current configuration does or does not satisfy the users' needs. Second, the treatment layer of the goal model must contain all and only the actions applicable in the context. This is very important for avoiding the system to be driven in undesired configurations. This depends on a proper definition of the rules for where and how it is possible to move data from a location to another set by the data administrator. Finally, impacts linking the treatments to goals must properly represent the effects of enacting the selected action. For this, the expert's knowledge is very relevant, but might not be enough. To help in the definition and refinement of the impacts, real effects are analyzed to improve the model by updating impacts according to what observed at run-time. Correct impacts enable us to predict the positive and negative effects of an action and to avoid disruptive decisions. A sound model thus provides all the elements to detect and react to violations taking informed decisions.

The described framework considers only the goal model for a specific user at a time. In future work, we will consider multi-users scenarios, where multiple goal models, potentially defining conflicting requirements, will be evaluated. In this case, two solutions can be adopted: (i) a centralized decision system that will have the control on all the goal models and it will select the best adaptation action and (ii) a distributed decision system that will divide the responsibility for the selection of the best adaptation actions among all its participants.

6.4. Using the Goal-Model in the Healthcare Scenario

Referring to our motivating example discussed in section 3 and to the goal model shown in **Figure 5**, we explain the usage of the proposed approach through some examples.

Let us suppose that the telecommunication provider of user A's smartphone is experiencing traffic congestion. This negatively impacts both latency and availability, which decrease to the point that they violate both the *Fast response* and the *High accessibility* goals. Consequently, the goal model can be used to find compensating actions capable of fulfilling the requirements once again. Based on the goal model, the adaptation action M_{CE} (i.e., move data from cloud to edge) is selected, as it has a positive impact on both violated goals, and the negative impact on other goals is negligible. To decrease the time required to perform data movement, the *Data reduction* transformation is additionally applied while moving data from the cloud to user B smartphone. Although *Data reduction* has a negative impact on the *Error Free* goal, the effect of such impact is not sufficient to violate that goal. Its execution thus has a positive effect.

Let us now consider that the volume of UV light measurements on user A's smartphone exceeds 10,000 samples, violating the *Light client* goal. Based on the goal model, the adaptation actions labeled M_{EC} and M_{EE} are selected, as they both have a positive impact on the *Light client* goal. In terms of the smartphone of user B in proximity with the one of user A, the adaptation action M_{EE}, which moves data between the two devices, is enacted, as it has no negative effects on the other goals, and its cost is lower or equal to the one of every action of the M_{EC} class. .

Let us also consider that it is possible to experience issues related to the reliability of values received from the UV sensor. Such problems can be caused by two main reasons: communication problems between sensors and the smartphone and degradation of the sensor's performance due to, for example, low battery or failures. In both cases, the reliability of the UV sensor values decreases to the point that it no longer fulfills the completeness and/or accuracy requirements specified for the *Error free* goal. Consequently, the adaptation actions labeled M_{EC} and *DataCleaning* are identified as candidates. Since M_{EC} has a negative impact on the *Fast response*, *High accessibility*, and *Privacy* goals, it is set aside in favor of *DataCleaning*, which will have only impact on *Fast Response* . In fact, enabling the data cleaning transformation will, on the one hand, take longer to process and display data but, on the other hand, it will try to provide reliable results. Within the "inputting values" features, null values will be detected and (if possible) substituted with reliable values. The "outlier detection" will analyze outliers, that will be removed or substituted with acceptable values if related to data glitches. In any case, it is necessary to underline that, if the quantity of values received is too low, no cleaning operation is possible, and the application should warn users of the system failure.

7. TOOL EVALUATION

The main feature of the software component we developed consists of deciding which is the best adaptation action to be enacted and, consequently, foreseeing the effects of such actions. We therefore executed tests to measure the ability of the software component to perform a choice that leads the system to a configuration that does not violate any goals defined in the goal model.

We simulated typical configurations for the case study and triggered several violations multiple times. We measured how many times the system is brought to a configuration where violations are removed and how many actions are enacted to reach such configuration.

We repeated the test with (i) a growing number of violations, (ii) reduced the number of edge/cloud resources where it is possible to move data, and (iii) reducing the quality of the network connection between resources. We have implemented software optimizations that allow the analysis of available resources and the selection of the best one; however, these optimizations cannot be applied on a network where, especially in a fog environment, the connection may not be stable. We therefore ran an additional set of tests to verify the behavior of the software component when optimizations cannot be applied.

Figure 6 shows the number of adaptation actions (Y axis) enacted based on the number of violations detected (X axis) simultaneously. The dashed line shows the situations in which multiple goal models (one for each user/application) are managed. In the test, increasing the number of violations corresponds to the introduction of an additional goal model; at any step, therefore, only one violation per goal model is detected. For example, five violations mean that five goal models (one for each user/application) detected one violation each. In this case, the number of adaptation actions, required to reach a system without violations, is equal to the number of violations received. The software examines one violation at a time and enacts the corresponding adaptation action to solve it. The solid line, instead, shows the behavior with multiple violations on a single goal model. As can be seen, in the experiments one adaptive action is sufficient to resolve all violations. By comparing the two behaviors, we can observe that, for a single goal model, an action can resolve multiple violations. Actions, however, have an impact on the system only at the local level. When multiple goal models are considered, an action must be taken for each goal model that has detected a violation.

Figure 7 shows the number of adaptation actions necessary in the event of deterioration of the quality of the network. We simulated the network using virtual connections, each of them with a set of properties, such as latency. The chart in **Figure 7** shows in the X axis the number of virtual network connections that can be used to restore a configuration of the system without violations, while the Y axis represents the number of adaptation

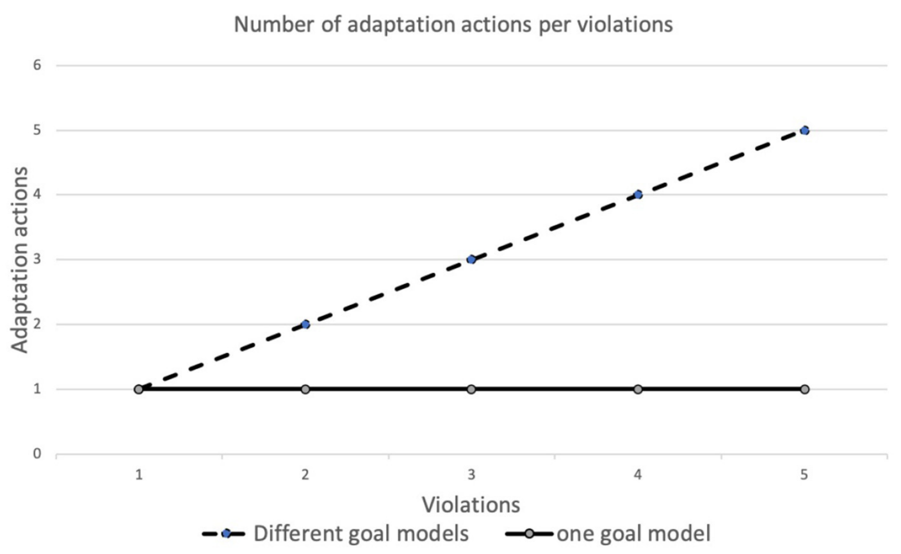

FIGURE 6 | Number of adaptation actions per violation.

actions enacted to bring the system to a configuration where no violations are detected. For a large number of available network connections (3, 4, 5) the correct decision is made immediately. For fewer available network connections, the correct decision is made after the second adaptation action. We repeated the experiment several times, but the number of adaptation actions was always the same since the software component uses a deterministic algorithm for the decision.

Summarizing what is shown in this section, the results of the experiments show that the number of violations affects the selection of the best adaptation action. When we deal with a single goal model (solid line in **Figure 6**), a single adaptation action can solve all violations. Instead, with multiple goal models (dashed line in **Figure 6**) each violation is triggered by a different node, therefore, the number of adaptation actions needed is equal to the number of violations detected. Concerning the relation between the number of adaptation actions and the quality of the network, the results in **Figure 7** have shown that the lower the network quality, the higher the number of adaptation actions that may be needed to restore a configuration of the system with no violations. Network connections in fog environments are unstable and their characteristics change frequently. To know the state of the network, a continuous benchmark would be necessary; however, the impact of this activity would create an excessive overload. The software component therefore tries to implement an adaptation action and waits for the next violation.

8. RELATED WORK

The evolution of data management systems in the last year has confirmed that the "one size fits all" approach is no longer valid (Stonebraker and Cetintemel, 2005) and this is also confirmed in the healthcare domain. In fact, nowadays, data intensive applications (Kleppmann, 2017) are not based on a unique database technology (e.g., relational databases) (Prasad and Sha, 2013). Also, the computation is now polyglot (Kaur and Rani, 2015), i.e., different modules are developed with different languages. This trend has been boosted also by the availability of platforms that usually support the micro service architectural style.

Although these new approaches provide a support for an easy development and execution of scalable and reliable solutions, the negative aspect concerns the need for inter-process communications in place of the shared memory access that is heavily affected by the network performances (Dragoni et al., 2017). For this reason, proper data management is required, and the information logistics principles are useful in this context (Sandkuhl, 2008; Haftor et al., 2011). In particular, (Michelberger et al., 2013) identifies different perspectives around which Information Logistics can be studied: e.g., from an organizational standpoint in terms of how to exploit the data collected and managed inside an organization for strategy purposes or how to properly distribute the data in a supply chain management. The so-called user-oriented Information Logistics (i.e., the delivery of information at the right time, place, and with the right quality and format to the user) advocates data movement (D'Andria et al., 2015). The issue of inter-process communication has been faced also in Vitali and Pernici (2016) based on healthcare scenario. In this work, hidden dependencies between the processes of different organizations where discovered by taking advantage of the data collected by the IoT devices in the environment. By combining and analyzing the information generated by different actors, an improved coordination between stakeholders can thus be reached. The issue of how to collect and manage these data remains open.

The approach proposed in this paper to express requirements about data movement relies on goal-based models that are usually

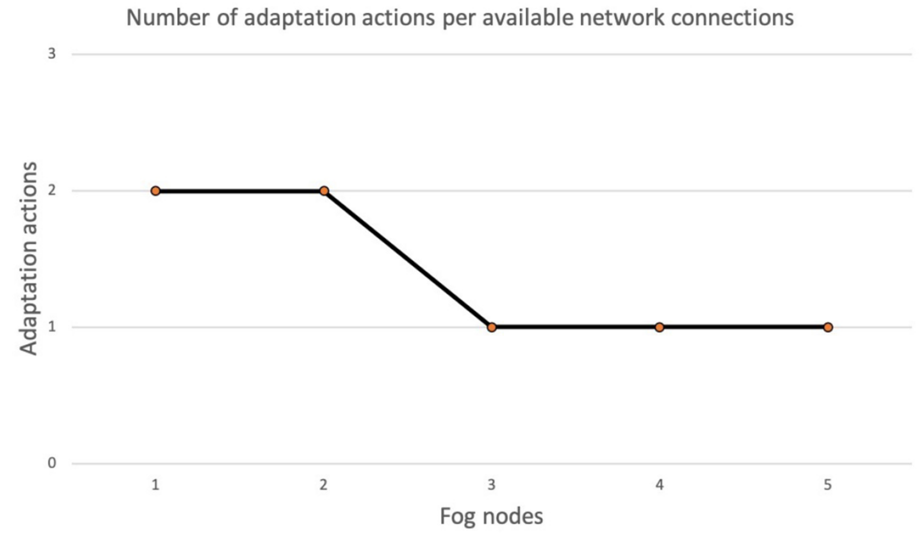

FIGURE 7 | Number of adaptation actions per available network connection.

adopted in requirement engineering to specify the objectives of users and applications to be designed (Van Lamsweerde, 2001; Amyot and Mussbacher, 2011; Horkoff et al., 2015). By using the tree-like structures of goal models, decisions on which subset of the modeled goals must be achieved can be taken. To this aim, several techniques have been proposed (Letier and Van Lamsweerde, 2004; Horkoff and Yu, 2016). The satisfaction analyzes propagate the satisfaction or denial of goals forward and backward in the goal tree structure. The forward propagation (Letier and Van Lamsweerde, 2004) can be used to check alternatives, while the backward propagation (Giorgini et al., 2003; Sebastiani et al., 2004; Chung et al., 2012) can be used to understand what are the consequences of a satisfied or denied goal.

Among the several requirements that can be expressed through our application of the goal-based model, the quality of data and quality of service aspects are the most relevant ones; in this paper, they are considered together under the data utility umbrella. Data utility has been defined in different ways in the literature. In statistics (i), it has been defined as "A summary term describing the value of a given data release as an analytical resource. This comprises the data's analytical completeness and its analytical validity" (Hundepool et al., 2012). In business (ii), it has been defined as "business value attributed to data within specific usage contexts" (Syed et al., 2008). In IT environments (iii), it has been defined as as "The relevance of a piece of information to the context it refers to and how much it differs from other similar pieces of information and contributes to reduce uncertainty" (Kock and Kock, 2007). More related to a Fog computing environment, (Cappiello et al., 2017) defines data utility as a numeric measure that reflects the relative importance and value contribution of a record from a business/usage perspective and provides a flexible approach that has been adopted in this paper to cover different types of applications as well as customizable set of data quality parameters. In fact, in the literature, some papers consider data utility in specific usage contexts (Ives et al., 1983; Lin et al., 2015; Wang et al., 2016) or on a specific set of data quality dimensions, e.g., accuracy, accessibility, completeness, currency, reliability, timeliness, and usability (Skyrme, 1994; Moody and Walsh, 1999).

9. CONCLUDING REMARKS

The ever-growing adoption of IoT-based solutions in the healthcare sector has resulted in a significant increase in data production, which could have the potential to be used in internal hospital processes but could also be relevant externally. On this basis, this document presented an approach based on the Fog computing paradigm, which demonstrates how this paradigm fits perfectly as a way to organize a distributed software solution in which data is produced at the edge of the network and consumed in other nodes that could be internal or external while preserving the data utility requirements. This goal was achieved by considering a formalization of data utility defined as a combination of data quality and quality of service. In addition, a goal-based model approach is adopted to select and enact an adaptation action capable of recovering the situation in the event that data utility is not satisfied.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by Ospedale San Raffaele Ethical Committee

(Protocollo ISEE - 720571-1 Title: Use of sensorized smartglasses for user's context and activity awareness: a validation study). The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

This work was supported by European Commission H2020 Programme through the DITAS Project no. 731945. This work expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this work.

REFERENCES

Ahmed, E., Yaqoob, I., Gani, A., Imran, M., and Guizani, M. (2016). Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges. *IEEE Wireless Commun*. 23, 10–16. doi: 10.1109/MWC.2016.7721736

Alaa, M., Zaidan, A., Zaidan, B., Talal, M., and Kiah, M. (2017). A review of smart home applications based on Internet of Things. *J. Netw. Comput. Appl*. 97, 48–65. doi: 10.1016/j.jnca.2017.08.017

Amyot, D., and Mussbacher, G. (2011). User requirements notation: the first ten years, the next ten years. *J. Softw*. 6, 747–768. doi: 10.4304/jsw.6.5.747-768

Atzori, L., Iera, A., and Morabito, G. (2010). The Internet of Things: a survey. *Comput. Netw*. 54, 2787–2805. doi: 10.1016/j.comnet.2010.05.010

Batini, C., and Scannapieco, M. (2016). *Data and Information Quality - Dimensions, Principles and Techniques. Data-Centric Systems and Applications*. Springer. doi: 10.1007/978-3-319-24106-7_15

Bermbach, D., Pallas, F., Pérez, D. G., Plebani, P., Anderson, M., Kat, R., et al. (2018). "A research perspective on fog computing," in *Service-Oriented Computing-ICSOC 2017 Workshops*, eds L. Braubach, J. M. Murillo, N. Kaviani, M. Lama, L. Burgueno, N. Moha, and M. Oriol (Cham: Springer International Publishing), 198–210. doi: 10.1007/978-3-319-91764-1_16

Cappiello, C., Pernici, B., Plebani, P., and Vitali, M. (2017). "Utility-driven data management for data-intensive applications in fog environments," in *Advances in Conceptual Modeling*, eds S. de Cesare and U. Frank (Cham: Springer International Publishing), 216–226. doi: 10.1007/978-3-319-70625-2_20

Chung, L., Nixon, B. A., Yu, E., and Mylopoulos, J. (2012). *Non-functional Requirements in Software Engineering, Vol. 5*. Springer Science & Business Media.

D'Andria, F., Field, D., Kopaneli, A., Kousiouris, G., Garcia-Perez, D., Pernici, B., et al. (2015). "Data movement in the Internet of Things domain," in *European Conference on Service-Oriented and Cloud Computing* (Taormina: Springer), 243–252. doi: 10.1007/978-3-319-24072-5_17

Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., et al. (eds.). (2017). "Microservices: yesterday, today, and tomorrow," in *Present and Ulterior Software Engineering* (Cham: Springer), 195–216. doi: 10.1007/978-3-319-67425-4_12

Ducato, R. (2016). "Cloud computing for s-health and the data protection challenge: getting ready for the general data protection regulation," in *IEEE International Smart Cities Conference, ISC2 2016* (Trento), 1–4. doi: 10.1109/ISC2.2016.7580803

Giorgini, P., Mylopoulos, J., Nicchiarelli, E., and Sebastiani, R. (2003). Formal reasoning techniques for goal models. *J. Data Semant*. 1, 1–20. doi: 10.1007/978-3-540-39733-5_1

Haftor, D. M., Kajtazi, M., and Mirijamdotter, A. (2011). "A review of information logistics research publications," in *International Conference on Business Information Systems* (Poznan: Springer), 244–255. doi: 10.1007/978-3-642-25370-6_24

Horkoff, J., Borgida, A., Mylopoulos, J., Barone, D., Jiang, L., Yu, E., et al. (2012). "Making data meaningful: the business intelligence model and its formal semantics in description logics," in *Proc. of on the Move to Meaningful Internet Systems* (Rome), 700–717. doi: 10.1007/978-3-642-33615-7_17

Horkoff, J., Li, T., Li, F.-L., Salnitri, M., Cardoso, E., Giorgini, P., et al. (2015). Using goal models downstream: a systematic roadmap and literature review. *Int. J. Inform. Syst. Model. Design* 6, 1–42. doi: 10.4018/IJISMD.2015040101

Horkoff, J., and Yu, E. (2016). Interactive goal model analysis for early requirements engineering. *Req. Eng*. 21, 29–61. doi: 10.1007/s00766-014-0209-8

Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E. S., Spicer, K., et al. (2012). *Statistical Disclosure Control*. John Wiley & Sons. doi: 10.1002/9781118348239

IEEE (2018). *IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing*. IEEE Std 1934-2018.

ISO/IEC (2010). *ISO/IEC 25010 System and Software Quality Models*. Technical report.

Ives, B., Olson, M. H., and Baroudi, J. J. (1983). The measurement of user information satisfaction. *Commun. ACM* 26, 785–793. doi: 10.1145/358413.358430

Kaur, K., and Rani, R. (2015). Managing data in healthcare information systems: many models, one solution. *Computer* 48, 52–59. doi: 10.1109/MC.2015.77

Klein, A., and Lehner, W. (2009). Representing data quality in sensor data streaming environments. *J. Data Inform. Qual*. 1:10:1–10:28. doi: 10.1145/1577840.1577845

Kleppmann, M. (2017). *Designing Data-Intensive Applications*. Beijing: O'Reilly.

Kock, N., and Kock, N. (2007). *Encyclopedia of E-collaboration. Information Science Reference*. Hershey, PA: IGI Publishing.

Letier, E., and Van Lamsweerde, A. (2004). "Reasoning about partial goal satisfactionfor requirements and design engineering," in *ACM SIGSOFT Soft. Eng. Notes*, Vol. 29 (Newport Beach, CA: Association for Computing Machinery), 53–62. doi: 10.1145/1041685.1029905

Lin, Y. C., Wu, C.-W., and Tseng, V. S. (2015). *Mining High Utility Itemsets in Big Data*. Cham: Springer International Publishing. doi: 10.1007/978-3-319-18032-8_51

Michelberger, B., Andris, R.-J., Girit, H., and Mutschler, B. (2013). "A literature survey on information logistics," in *International Conference on Business Information Systems* (Poznan: Springer), 138–150. doi: 10.1007/978-3-642-38366-3_12

Moody, D., and Walsh, P. (1999). "Measuring the value of information: an asset valuation approach," in *European Conference on Information Systems* (Copenhagen), 496–512.

Peng, X., and Pernici, B. (2016). "Correlation-model-based reduction of monitoring data in data centers," in *2016 5th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)* (Rome: IEEE), 1–11. doi: 10.5220/0005794803950405

Plebani, P., Garcia-Perez, D., Anderson, M., Bermbach, D., Cappiello, C., Kat, R. I., et al. (2017). Information logistics and fog computing: the DITAS approach. *CEUR Workshop Proc*. 1848, 129–136.

Plebani, P., Salnitri, M., and Vitali, M. (2018). "Fog computing and data as a service: a goal-based modeling approach to enable effective data movements," in *International Conference on Advanced Information Systems Engineering* (Tallin: Springer), 203–219. doi: 10.1007/978-3-319-91563-0_13

Prasad, S., and Sha, M. S. N. (2013). "Nextgen data persistence pattern in healthcare: polyglot persistence," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)* (Tiruchengode), 1–8. doi: 10.1109/ICCCNT.2013.6726734

Redman, T. C. (1996). *Data Quality for the Information Age*. Norwood, MA: Artech House.

Sandkuhl, K. (2008). *Information Logistics in Networked Organizations: Selected Concepts and Applications*. Berlin; Heidelberg: Springer. doi: 10.1007/978-3-540-88710-2_4

Satyanarayanan, M., Bahl, P., Caceres, R., and Davies, N. (2009). The case for VM-based cloudlets in mobile computing. *IEEE Pervas. Comput*. 8, 14–23. doi: 10.1109/MPRV.2009.82

Sebastiani, R., Giorgini, P., and Mylopoulos, J. (2004). "Simple and minimum-cost satisfiability for goal models," in *Proc. of Int. Conference on Advanced Information Systems Engineering* (Berlin; Heidelberg: Springer), 20–35. doi: 10.1007/978-3-540-25975-6_4

Skyrme, D. J. (1994). Ten ways to add value to your business. *Manag. Inform*. 1, 20–25.

Stonebraker, M., and Cetintemel, U. (2005). "One size fits all":an idea whose time has come and gone," in *21st International Conference on Data Engineering (ICDE'05)*, 2–11. doi: 10.1109/ICDE.2005.1

Syed, M. R., Syed, S. N., Syed, M. R., and Syed, S. N. (2008). *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications. Information Science Reference*. Hershey, PA: IGI Publishing. doi: 10.4018/978-1-59904-887-1

Trihinas, D., Pallis, G., and Dikaiakos, M. (2018). Low-cost adaptive monitoring techniques for the Internet of Things. *IEEE Trans. Serv. Comput*. 1, 1. doi: 10.1109/TSC.2018.2808956

Van Lamsweerde, A. (2001). "Goal-oriented requirements engineering: a guided tour," in *Proceedings Fifth IEEEInternational Symposium on Requirements Engineering* (Toronto, ON: IEEE), 249–262.

Varshney, P., and Simmhan, Y. (2017). "Demystifying fog computing: characterizing architectures, applications and abstractions," in *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)* (Madrid: IEEE), 115–124. doi: 10.1109/ICFEC.2017.20

Vitali, M., and Pernici, B. (2016). "Interconnecting processes through IoT in a health-care scenario," in *IEEE International Smart Cities Conference (ISC2)* (Trento: IEEE), 1–6. doi: 10.1109/ISC2.2016.7580760

Vitali, M., Pernici, B., and O'Reilly, U. (2015). Learning a goal-oriented model for energy efficient adaptive applications in data centers. *Inf. Sci*. 319, 152–170. doi: 10.1016/j.ins.2015.01.023

Wang, J., Zhu, X., Bao, W., and Liu, L. (2016). "A utility-aware approach to redundant data upload in cooperative mobile cloud," in *9th IEEE Int'l Conf. on Cloud Computing, CLOUD 2016* (San Francisco, CA), 384–391. doi: 10.1109/CLOUD.2016.0058

Wang, R. Y., and Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *J. Manage. Inform. Syst*. 12, 5–33. doi: 10.1080/07421222.1996.11518099

Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet Things J*. 1, 22–32. doi: 10.1109/JIOT.2014.2306328

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: frontiersin.org/about/contact

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership