future internet

Article

A Virtual Learning Architecture Enhanced by Fog Computing and Big Data Streams

Riccardo Pecori [1,2]

[1] SMARTEST Research Centre, eCampus University, Via Isimbardi 10, 22060 Novedrate, CO, Italy; riccardo.pecori@uniecampus.it

[2] Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze 181/A, 43124 Parma, PR, Italy; riccardo.pecori@unipr.it

Received: 19 November 2017; Accepted: 26 December 2017; Published: 3 January 2018

Abstract: In recent years, virtual learning environments are gaining more and more momentum, considering both the technologies deployed in their support and the sheer number of terminals directly or indirectly interacting with them. This essentially means that every day, more and more smart devices play an active role in this exemplary Web of Things scenario. This digital revolution, affecting education, appears clearly intertwined with the earliest forecasts of the Internet of Things, envisioning around 50 billions heterogeneous devices and gadgets to be active by 2020, considering also the deployment of the fog computing paradigm, which moves part of the computational power to the edge of the network. Moreover, these interconnected objects are expected to produce more and more significant streams of data, themselves generated at unprecedented rates, sometimes to be analyzed almost in real time. Concerning educational environments, this translates to a new type of big data stream, which can be labeled as educational big data streams. Here, pieces of information coming from different sources (such as communications between students and instructors, as well as students' tests, etc.) require accurate analysis and mining techniques in order to retrieve fruitful and well-timed insights from them. This article presents an overview of the current state of the art of virtual learning environments and their limitations; then, it explains the main ideas behind the paradigms of big data streams and of fog computing, in order to introduce an e-learning architecture integrating both of them. Such an action aims to enhance the ability of virtual learning environments to be closer to the needs of all the actors in an educational scenario, as demonstrated by a preliminary implementation of the envisioned architecture. We believe that the proposed big stream and fog-based educational framework may pave the way towards a better understanding of students' educational behaviors and foster new research directions in the field.

Keywords: big data; big data streams; fog computing; e-learning; virtual learning environment; Internet of Things

1. Introduction

The forecast of a worldwide network following the Internet of Everywhere (IoE) paradigm is currently becoming a reality, mainly thanks to those devices called Smart Objects (SOs). They surely are a part of the Internet of Things (IoT) vision, and they are represented by sensors, smartphones, wearables, tablets and all other equipment allowing for new forms of interaction between the physical world, the things and the people to whom they are attached. Reliable estimates foresee that, by 2020, billions of SOs are expected to be deployed in urban, home, industrial and rural scenarios, in order to collect relevant data, which may be used to build and foster new applications and innovative services for all the involved stakeholders (from citizens to institutions). This huge amount of information, coming from heterogeneous sources, can be framed in the big data paradigm, characterized in terms of volume, velocity, variety, veracity and value [1]. Indeed, the most recent big data techniques, which are so popular today mainly due to the spreading of those online and social services they claim to support, attempt at processing extremely large amounts of heterogeneous data for multiple purposes. As a matter of fact, the majority of these techniques has been mainly conceived of to deal with extremely large volumes of data, hence focusing on their amount, rather than on their real-time processing and dispatching.

To fill this gap, a new paradigm, alternatively labeled as big data streams or big stream [2], is emerging within those parts of the research community more involved in IoT data traffic. Indeed, this model is more interested in those aspects that, despite being as important as volume, seem to have been somehow forgotten by traditional big data solutions: the data generation rate, the real-time and low latency requirements some consumers have, as well as the management of data flows.

Besides this, another concept, namely Fog Computing (FC) [3], is quickly arising as the perfect companion of big data streams: as a matter of fact, the goal of this emerging paradigm is to reduce the amount of data that needs to be transported to the cloud for processing, analysis and storage. Such a technology improves the overall efficiency and real-timeliness, by migrating computing resources, applications and services to the edges of the network.

At the time of writing, several educational institutions are relying more and more on the usage of the so-called Virtual Learning Environments (VLEs), mainly fostered by the evolution in the communication bandwidth and in the increased availability (at paltry costs) of the aforementioned pervasive devices, themselves able to provide a connection anywhere and anytime. Thus, both traditional and e-learning-based universities are evolving towards virtual and mobile learning (m-learning) frameworks, thanks to the IoT social revolution and, in the very near future, towards scenarios heavily characterized by the integration of fog and cloud technologies.

As a direct consequence of the adoption of these technologies, educational big data mining is emerging as a novel research scenario [4], wherein big data and big data stream mining techniques may be applied to the valuable information coming from different educational data sources, be they either sensor-like gadgets (e.g., wearables) or full-fledged devices (e.g., tablets and laptops). Moreover, these new scenarios allow one to offer a partial solution to the most common legacy issues connected with distance learning environments and educational data mining techniques, such as the requirement for: (i) increasing *scalability* (considering both infrastructure and algorithms); (ii) *real-time responsiveness* (for quick responses towards students' requirements and needs); and (iii) *adaptivity* (since workload peaks may rapidly follow more quiet periods). In the light of all said and to cope with the increased demand of scalable and adaptive analytics, this article presents a layered functional architecture enabling time-efficient analysis of educational big data streams through an effective usage of big data stream and fog computing technologies combined together. The envisioned architecture has been implemented in a preliminary version and has provided some good results in thwarting dropouts of students of a distance learning university.

The remainder of this paper is structured as follows. First, we perform a review of the state of the art about the current virtual learning environment solutions and related open issues, then we present the main features and characteristics of big data streams and fog computing, highlighting their main advantages and drawbacks. Subsequently, we provide a possible overall structure of an enhanced e-learning architecture based on these techniques, describing in detail the involved layers and also a possible practical, even if preliminary, implementation and a case study. Finally, we offer an overview of the possible advantages of big data streams and fog computing techniques applied to a traditional distance learning scenario, and we corroborate the proposed vision with some useful conclusions, also tracking the main future research lines we envision in this field, currently still in flux.

2. Distance Learning

E-learning is connected with virtualized distance learning through electronic communication devices and mechanisms, with their own functionalities both as a support and as an integral part in the teaching-learning procedures. Virtual courses, supported partly or in full by the distance learning approach, rely on web-based architectures, themselves capable of storing and delivering the so-called Learning Objects (LOs), i.e., various kinds of educational materials (slides, videos, quizzes, tests, etc.) having a single educational objective. LOs are usually encoded into container formats, such as SCORM (Shareable Content Object Reference Model), themselves capable of recording the time spent by the students on the LO itself. These items are part of a so-called Virtual Learning Environment (VLE), a digital framework and web-based portal made of all learning sources, with which students and instructors can interact in a constructivist way. Moodle (http://moodle.org/) (Modular Object-Oriented Dynamic Learning Environment) is surely one of the most renowned learning environments, especially thanks to its modularity, its reusability in different contexts and for the constructionist pedagogical principles on which it is based. Concerning the data sources of a typical VLE, they are usually multifaceted and consist of four main layers [4]:

- VLE-specific sources;
- general application sources;
- middleware sources;
- systems software sources.

Some examples of the most widespread sources are the following: a user profile, private and shared annotation spaces, SCORM containers, chats, forums, blogs, tests and e-portfolios, wikis, concept maps, and the like.

Distance learning has a clear advantage compared with classical attendance groups: the higher number of students that can have access to educational contents. This is achieved, for example, through the deployment of MOOC (Massive Open On-line Course) platforms, such as the ones provided by the Khan Academy (http://www.khanacademy.org), the EduOpen(http://en.eduopen.org/) project, the MIT Open CourseWare (http://ocw.mit.edu/index.htm), Coursera (http://www.coursera.org/), edX (http://www.edx.org/), and the like. However, the advantages of e-learning systems are not limited to the aforementioned, but are multifaceted and can address different stakeholders: students, instructors, IT staff, institutions and educational researchers [5].

2.1. State of the Art

After introducing distance learning and VLEs, in this subsection, we review the current state of the art in order to point out the advantages, as well as the limitations of present solutions that led us to devise a novel architecture. This review is performed according to the different stakeholders mentioned beforehand.

2.1.1. Students

Students usually enjoy VLEs for their ease of usage and communication, the presence of a friendly environment, the ubiquity allowed in the fruition of LOs, the possibility of cooperation and collaboration among students and between instructors and students themselves [6], the availability of on-line tutoring services [7], the creation of personalized learning paths, as well as of communities and groups of peers. Moreover, the contribution in [8] witnesses that students that use VLEs achieve better performances than non-users: the beneficial effects of distance learning are confirmed both by online performances and final exam outcomes. Besides, the success of VLEs is measured by the students involved, according to their usage intentions, their actual use, satisfaction and benefits in employing it [9]. However, these successful conclusions are only true under certain circumstances, i.e., a minimum effort in using the VLE is required. Moreover, the contribution in [10] points out the fact that the usage of only VLEs, on the part of the students, may lead to excessive workload and early dropouts or to the adoption of some wrong strategies.

Some important characteristics that students search in modern VLEs are authenticity and reification [11]. These may regard the context of learning, specific learning tasks, the impact of the attended courses, but also the personal sphere, as well as the value of the courses as the students perceive them. LOs that allow a practical experimentation, in real life, of the knowledge they have learned are very appreciated by the students.

The solutions based on the cloud computing paradigm [5] partly fulfill these requirements for online learning applications. They also foster communications and resource-sharing between the students themselves and between the students and the teachers. Cloud-based VLEs allow also for the necessary flexibility to follow dynamic personalized learning paths, as well as to participate in complex laboratory environments and for the needed scalability that guarantees a stable quality of service. Moreover, cloud-based laboratories provide actualization and re-creation of real experiments, which result in being very important for a complete involvement on the students' part. Nevertheless, they are rather complicated to enact, especially if interactive or monitoring procedures are employed, rather than simple batch processes. Moreover, the cloud may present some risks for students if their personal data are not correctly handled and stored, especially if they are managed by external organizations, unknown to the students and not complying with privacy laws. Besides, not all cloud services provide a high enough level of performance, i.e., a sustainable connection and small delays and jitters, for all LOs, and this could affect interactive and collaborative learning tasks.

Despite these challenges, cloud-based virtual laboratories are still present in the recent literature. For example, in [12], the authors propose to engage students in practical activities exploiting their own mobile devices. This led to various advantages for them, such as ubiquitous and context-aware experiences in informal learning situations and the possibility to join group discussions anytime and anywhere. These, in turn, fostered problem-based learning and hands-on practical measurements and, at the same time, the motivation of the students to participate in the lab activities.

The importance of mobile-based learning (m-learning) for students is highlighted also in [13], where the authors present a possible app for integrating a Learning Management System (LMS) into mobile devices, also showing how this could increase the usage of distance learning on the part of the students, especially thanks to interactive tools such as forums, reminders and message systems. In [14], m-learning is successfully extended to ubiquitous learning. This regards the inclusion of students' everyday environments into the learning scenario, as well as the integration, into a unique framework, of daily learning collaborators, contents and services and digital augmentation of the physical objects of the students' real life.

2.1.2. Teachers

The main advantages of VLEs, when instructors are concerned, involve the flexibility of usage, the cooperation and collaboration among instructors and between instructors and students, the presence of common areas of knowledge to be shared across colleagues, the monitoring of students' progress, as well as creating and managing groups of students. However, the contribution in [10] alerts instructors and tutors to carefully follow students in the VLE usage, in order for them to avoid the adoption of the wrong strategies or bad habits, such as procrastination and trial and error. As a consequence, the employment of distance learning should be carefully tuned by teachers and adapted to the specific learning situations.

Moreover, the quality of the provided content, in a VLE, is an essential point for its successful employment, and this is totally the province of teachers [9]. In this regard, the reification of subjects is a very important characteristic of any online course, and the teachers are encouraged to provide also experiences close to the real life of students and the most practical as possible, in order to be perceived as actually useful, genuine or authentic by the students themselves [11].

In [15], new software modules were built in order to enhance Moodle with recommending capabilities targeted at teachers and instructors. These integrated modules allow for speeding up and customizing the creation of the courses, by taking advantage of past social usages of LOs on the part of colleagues teaching the same subjects or having similar pedagogical inclinations. The proposal seemed quite successful both in speeding up the time of creation of online courses and in the appreciation of a set of instructors, mainly those with little or scarce technical skills. Another work where recommending systems are applied to virtual learning environments is [16]. In this contribution, the authors focus on teachers again, especially considering their ICT-related skills and profile. The evaluation results of the contribution show that recommendations are personalized according to the ICT competences of instructors with a great degree of granularity thanks to the usage of fuzzy depiction.

Carefully evaluating the integration between the cloud and VLEs from the perspective of teachers and instructors is very important as well [17]. This validation should take place also considering some theoretical pedagogical underpinnings, e.g., relatedness together with the inherent affordance of the system and how the functionalities provided by the VLE are chosen and used. However, the solutions based onto the cloud computing paradigm are proven to increase the availability of educational services, such as the sharing of documents between students and instructors and across teachers, as well as reusing collaborative pedagogic tools [5].

Moreover, in cloud-based remote laboratories, the possible educational strategies provided to teachers increase [18]. They may vary from simulations, to recreation, where past collected actual data are used, until actualization, where lab equipment is remotely driven through micro-controllers in real time. Nevertheless, cloud-based solutions need to guarantee adequate performances, especially when dealing with interactive and collaborative learning tasks, and this is not always true.

Focusing on the context, Gomez et al. [19] propose a system, very useful for teachers, to engage students in personalized learning paths, as well as to monitor their achievements, by integrating the learning process into the students' own environment by means of their own mobile devices. In [14], ubiquitous learning is presented as an extension of mobile-based learning. In this scenario, teachers are helped with performing a better orchestration of learning strategies and to continuously adapt them to the outcomes of the students.

2.1.3. Institutions

What moved institutions towards VLEs is their flexibility of employment, the repeatability and the consistent control they provide, the reduction of costs (in both software and hardware), the possibility of monitoring students' progress, as well as the chance to manage groups of students and of communities.

Concerning saving costs, an important and wide-spread solution for VLEs concerns the cloud computing paradigm [5]. Such a technological choice allows for a centralized virtualization and re-use of low-performing legacy computers, as well as deploying applications without the need to buy many licenses. Moreover, some cloud-based learning environments have been devised ad hoc for virtual laboratories in different fields [18,20], thus allowing institutions, after specific agreements, to exploit laboratories of other universities or research centers. Nevertheless, the cloud could raise some issues, such as the inability to implement a pay-per-use scheme and open-source solutions or what is called vendor lock-in. The latter regards the lack of interoperability that could block institutions from sharing data and services across different clouds or dynamically porting them from one service provider to another.

2.1.4. IT Staff

As is the case with any IT-based system, distance learning environments must undergo a proper development, implementation and continuous improvement as well, usually performed by dedicated IT staff. As analyzed deeply in [9], the main design characteristics of a satisfactory VLE can be classified according to either the granularity of the provided features (coarse, medium or fine) or the origin of the features themselves. In this second case, the IT staff is in charge of those system features depending on the technical design of the environment such as trustworthiness, communication abilities, presence of feedback or of interactivity elements, media synchronicity, flexibility, screen design, accessibility and adaptability.

In this regard, in [21], a novel design approach of VLEs is described. The proposed solution tries to smoothly apply aspect-oriented software development and multiagent programming to traditional object-oriented LMSs, by mapping objects to agents as useful joining points. In particular, the author separates the functionalities and concerns of a complex team-learning environment into different agents, each one in charge of different tasks, such as infrastructure interfaces, learning styles, learning tools, learning functions, queries, and so on. Even if the task was complex, the author partly succeeded in improving adaptability and re-usability, proven to be more and more important in VLEs in turn.

Some other solutions, mainly based on the cloud computing paradigm [5], stress the reduction in installation efforts and maintenance times (through virtualization), as well as the necessary flexibility to set up complex lab scenarios and to reuse pre-configured frameworks. The cloud allows also for scalability in managing services with variable demands; however, it can also be sensitive to reliability problems, such as the possibility of outage episodes, which may affect IT staff especially in certain periods, such as enrollment, exams and the like. Moreover, in the cloud, different integration layers should be considered, e.g., the management of infrastructures, software, resources, services, as well as applications [22]. Particular attention should be employed, by IT technicians, to those levels, providing the coupling between the aforementioned technologies and a unified user interface.

Despite these challenges, cloud solutions for VLEs are still very popular. For example, in [18], the authors propose a proprietary cloud-based virtual lab for mechatronics, enriched with a virtual meeting system, including communications with experts, a booking system for the physical equipment and a monitoring channel based on webcams. The proposed solution sheds light on the importance of carefully adapting, on the part of a dedicated IT staff, the different layers of the cloud to the specific laboratory, with particular care to micro-controllers, local servers, pieces of middleware, appropriate operating systems and devices (Arduino, Raspberry Pie, etc.). In [20], another system for virtual laboratories is presented. This solution takes advantage of remote access to virtual machines (Virtual Remote Labs) and needs IT staff to isolate virtual machines during certain time frames, when they have to be connected to actual physical devices or with network equipment. Technicians are also necessary to manage the four main layers (management, cloud, platform and data centers) of the proposal and to set hypervisors, such as VMware (http://www.vmware.com), and virtual infrastructure managers, such as Open Nebula (http://opennebula.org/). Moreover, some issues still to be solved concern the careful management of network communications and the need for a unique user interface and single sign on, as well as energy saving.

In [12], other forms of virtual labs are exploited thanks to the mobile devices of students. This solution, to be enabled by the IT staff, requires the delivering of free apps to be installed on the mobile devices involved, the management of a number of licenses at least equal to the number of students, the compatibility of the apps with different mobile operating systems, as well as with different sensors. Another mobile-learning solution is presented in [13], but it is based onto a client-server paradigm and presents some difficulties, to be overcome by the IT staff, such as the compatibility of the app with different devices and the accurate designing of the app according to different learning requirements and desired outcomes, for both students and teachers.

The contribution in [19] suggests the usage of Internet of Things devices and sensors in order to foster the motivation for students to employ the proposed system more than virtual worlds or immersive learning. Nevertheless, the proposed application presents some issues that IT technicians should solve: it is based on a traditional client-server paradigm that could be obsolete when facing new dynamic and demanding services, and the reasoner layer is critical and could be a single point of failure. In [14], ubiquitous learning is shown as an evolution of mobile learning. In this scenario, developers, who have to maintain and monitor the whole system at run time, are aided in selecting new algorithms should failures occur.

2.1.5. Researchers

The aspects of VLEs, in which researchers are usually interested, regard data mining techniques and the development or the discovery of novel educational patterns.

In [15], a recommending system for teachers is developed, and different learning object repositories were analyzed and tested. The results of a search query were listed according to both a basic relevance (based on a term frequency-inverse document frequency metric) and a local didactic relevance (based onto the relative position and the relative distance of other LOs employed in similar courses by colleagues with similar interests and ways of teaching). Nevertheless, one of the main limitation of this proposal regards the necessity of an opportune crawler to be adapted to different learning object repositories. In [16], another recommending system for teachers, integrated into a VLE, is presented. It is based both on the closeness in competences and on the abilities in using learning objects, while the instructor profile is composed of rating history, bookmark history, learning object access history and learning object creation history, combined together into an aggregation phase followed by a fuzzification step.

Another aspect, which is usually examined in terms of data analysis and educational research, is surely cloud computing [5], as it can support computations either for heavy simulation scenarios or for machine learning and data mining techniques. By taking advantage of the cloud paradigm, in [12], the authors propose wireless communicating virtual labs. The researchers' challenge, in this situation, is a careful verification of the validity of this model of virtual labs, which the authors claim to achieve better results compared to traditional models. As a matter of fact, caution must be employed towards virtual laboratories and other forms of mobile or ubiquitous learning. Researchers have to carefully analyze the factors fostering or hindering the adoption of such technologies, on the part of the students, according to the considered learning scenario [23]. Indeed, the individual background, or the environment students are from, the social pressure and personal psychological features may not often affect the adoption of mobile devices for learning as much as innovativeness, ease of use and perceived usefulness themselves, even if the age of students results sometimes in a limit for the smallness of the screens. Moreover, the positive impact of m-learning on the students' achievements must be evaluated very carefully as well, as from the contribution in [23], it results in being marginal, in spite of the chance to study anytime and anywhere. Taking always into consideration mobile and ubiquitous learning, Gomez et al. [19] propose a context-based system, to monitor the students' achievements, which leads to the learning process being in the students' own environment. The proposal is interesting for researchers, as it is based on ontologies capable of modeling the context the students are in, according to the location, to the student profile, to the time dimension and to the features of the learning activities. A reasoner software layer decides which activities and LOs are the most suitable for the specific students involved, according to their personal profile, as well as to time and location of the activities and of the students themselves.

In [14], ubiquitous learning is presented as a replacement for mobile learning. In this scenario, researchers, who are in charge of continuously evaluating the efficiency of the system both at a technical and at a pedagogical level, are fostered to discover new interesting learning patterns. This is achieved by introducing a novel meta-level in order to adaptively monitor and control actuators and perceptors and by adding introspection functionalities. Even so, this formal model is only theoretical, and its actual implementation, as well as its real effectiveness are still to be evaluated properly.

2.2. Some Still Unresolved Issues

As shown in previous subsections, different interesting proposals of VLEs are present in the literature with great advantages towards the main stakeholders. However, some issues are still experienced by traditional and legacy e-learning platforms, such as small scalability, especially at the infrastructural level, poor efficiency in the dynamic resource usage and workload scheduling, software licenses and hardware costs, maintenance time and expenditures, etc. This is why some cloud-based distance learning platforms are emerging [24], and the cloud, be it public, private or hybrid, is becoming a cornerstone of any modern VLE.

Nevertheless, some important research issues concerning the cloud and e-learning are still open [5]. They consider, for example, the way cloud infrastructures for education are built: currently, the Infrastructure as a Service (IaaS) layer is usually demanded from private middleware, while the Platform as a Service (PaaS) and Software as a Service (SaaS) layers are public services. Other challenges concern the need for easy-to-use scheduling and reservation functionalities, also in the presence of limited underlying private resources, the necessity of automatic scalability, according to the different periods of the year and of binding the monitoring metrics to the specific actions both at the IaaS level and at the PaaS and SaaS layers, the composition of complex learning modules, such as virtual laboratories and their integration in the cloud infrastructure, the interoperability between different learning services across different cloud standards, maybe in the form of Task as a Service (TaaS) or brokerage, and the support for mobile-learning capabilities such as ubiquity, synchronization, augmented reality, and so on. From the studied state of the art and the summarized open challenges, some needs emerge for improving current VLEs: the adaptation to particular learning situations, interactivity and real-timeliness, the closeness to reality, as well as authentic and real-world experiences, a trustworthy technical design, a layered adaptable and distributed architecture made of different modules for different granular analyses, careful attention to the coupling layers, more security, privacy and interoperability, maintenance of the traditional cloud layers and support for mobile learning and ubiquitous learning, as well as for context-aware and personal educational experiences.

In the following sections, we are going to introduce the basics of big data streams and fog computing that will enable us to propose an innovative e-learning architecture that can meet the aforementioned requirements.

3. Big Data Streams and Fog Computing

In this section, we summarize the fundamentals of big data streams and fog computing, in terms of the main concepts and current technologies. This part has the aim of furnishing a general background about the aforementioned topics, in order to fully understand the potentialities of the framework proposed in Section 4.

3.1. Big Data Streams

Big data streams are quickly becoming a major paradigm in the field of data science [25]. They arise in an ever-increasing range of fields—the web, sensors, intelligent systems, etc.—and are usually generated by modern platforms that involve a large number of applications, users and devices. Specific methodologies and technologies are required to deal with virtually infinite streams, non-stationary data and evolving pieces of knowledge.

Big data streams share some concepts and issues with the big data paradigm [26], such as the velocity, the volume and the variety. Indeed, big data and big data streams are characterized by a huge amount of information, generated at a fast rate, which may have an unknown structure. The main differences between big data and big data streams are related to the fact that the latter are produced at a faster rate, may be generated by several heterogeneous sources and need to be analyzed almost in real time. Indeed, big data streams provide temporally-ordered sequences of observations. These may or may not be time-stamped, and the time scale is variable, depending on the specific data source. However, the ordering of observations is significant because there may exist a meaningful inter-observation correlation (time series), or because the underlying phenomenon may be changing in time, either gradually (concept drift) or abruptly (concept shift). Each observation is made up of a complex record of data, itself built up by several variables of an unspecified nature, possibly obtained from heterogeneous streams and depending on the specific source.

During the last few years, a huge amount of paradigms and technologies has been introduced for handling, managing and analyzing big data [26]. These approaches are not suitable for handling big data streams in order to extract useful knowledge and insights from them. As a matter of fact, such approaches are typically based on batch processing and focus on the data themselves, rather than providing real-time processing and dispatching [27]. In order to deal with big data streams, we need technologies and methodologies for designing and implementing scalable and cost-competitive services, themselves capable of efficiently collecting and pre-processing data streams, in order to extract useful knowledge to exploit in a specific scenario. In the following, we summarize the main crucial issues that must be addressed when dealing with big data streams:

- The *data generation rate* is an important requirement that must be taken into consideration, along with the time needed to analyze the stream and to exploit its information [25];
- The infrastructure in charge of handling big data streams must be able to work in *real time* and to provide *low-latency responses*. These features are typically not taken into account by big data technologies, since data are statically stored, whereas when dealing with big data streams, information evolves and ages over time [28];
- *Data flows* must be accurately selected, pre-processed and analyzed for different purposes and final users. In particular, big data stream infrastructures might need to perform data aggregation, filtering or pruning, with different degrees of granularity, in order to minimize the latency in conveying the final computation output to certain users and to improve the accuracy of the responses they provide [29].
- The *knowledge extraction* from big data streams requires novel data mining approaches for working on-the-fly and with irregular, out-of-order and bursty inputs. Indeed, the models should be continuously updated in order to keep up with new developments, concept drifts, shifts, re-occurrences and the like [30].

In the last few years, a number of technologies and frameworks has been released for dealing with data streams that are quickly produced, change very quickly, are often incomplete, as well as imprecise and need a real-time analysis. Apache S4 (http://incubator.apache.org/s4/), Apache Storm (http://storm.apache.org/) and Apache Samza (http://samza.apache.org/) are some Apache projects focusing on efficiently collecting, handling and managing big data streams.

S4 is a Java-based platform, initially released by Yahoo!, capable of processing continuous streams of data in parallel. It is based on pluggable modules that can be combined in order to create complex processing systems. Storm is a distributed real-time computation system for quickly analyzing big streams of data. It is based on a master-slave approach and is composed by both a complex event processor and a distributed computation framework. Samza is a framework that processes stream messages as they arrive, one at a time. streams are divided into partitions that are in reality an ordered sequence of read-only messages. Samza relies on external facilities, for example Hadoop's YARN (http://hortonworks.com/apache/yarn/) (Yet Another Resource Negotiator) and Apache Kafka (http://kafka.apache.org/).

The aforementioned technologies are simply stream processing engines that can be used as a basis for data mining and machine learning frameworks in order to extract useful knowledge from big data streams. Indeed, recently, the SAMOA (http://samoa.incubator.apache.org) (Scalable Advanced Massive Online Analysis) framework has been released in order to run distributed machine learning algorithms on stream processing engines, such as the aforementioned S4, Storm and Samza. Actually, the number of frameworks that allow handling both static big data and big data streams is increasing day by day. Recently, Apache Flink (http://flink.apache.org/), which provides data distribution, communication and fault tolerance for distributed computations over data streams, including also a machine learning library, has been introduced. Furthermore, Apache Spark (http://spark.apache.org/) has its own module for handling data streaming and a library of machine learning algorithms, namely MLib (http://spark.apache.org/mllib/). To the best of our knowledge, *H_2O* (http://www.h2o.ai) is the most recent open-source framework that provides a parallel processing engine, analytics and machine learning libraries, along with data pre-processing and evaluation tools. Finally, where stream management is concerned, also the way data are stored must be considered, and this leads to considering the so-called Data Stream Management Systems (DSMS), capable of performing continuous queries over streams of data, being infinite in size and thus unbounded in length. Some examples thereof are InfoSphere (http://www-01.ibm.com/software/data/infosphere/streams/), SQLStream (http://sqlstream.com/feature/stream-processing-definition/), StreamBase (http://www.tibco.com/streaming-analytics), and the like.

3.2. Fog Computing

Big streams of data, coming from IoT devices, generate huge amounts of information, which can be subsequently processed and used to build several useful services for final users. The cloud, and its rack-mounted, warehouse-like data centers, made up the natural collection environment for big data, due to its scalability, robustness and cost-effectiveness. However, Figure 1 shows how recent architectures are moving towards a hierarchy of different levels involved in data collection, and among these, we can clearly distinguish the fog layer, usually populated with smart gateways being closer to sensors and smart devices than to cloud servers. Fog computing [31,32] is a recently-emerging paradigm, whose aim is to extend cloud elaboration and services to the edge of the network, by relying on the proximity to end-users, the dense geographical distribution and the support for mobility that IoT devices may guarantee, even in case of network connection unavailability. Fog specifically refers to IoT gateways, and these are indeed the so-called fog nodes, while edge computing considers the intelligence to be put further down, directly to the end devices. Both scenarios consider intelligent objects or devices equipped with proper software such as Cisco IOx (http://developer.cisco.com/site/iox/), Cisco Data In Motion (DMo) (http://developer.cisco.com/site/data-in-motion/discover/overview/), LocalGrid (http://www.localgridtech.com/) or PrismTech Vortex (http://www.prismtech.com/vortex). The wide geographical distribution and the moving of intelligence and computational power to the borders of the network make the fog computing paradigm particularly suited to real-time big data stream analytics, as presented in Section 3.1, both to foster new adaptive mining techniques while promoting analyses based on time and location information. In this perspective, the fog computing architecture is not meant to completely replace the cloud, but rather to complement it and interplay with it. As a matter of fact, fog computing corroborates cloud computing, and they both share the same resources, as well as many of the same mechanisms and attributes. The following innovative features are typical of the fog paradigm and enrich cloud-based architectures:

- the focus onto *very low and predictable latency*, e.g., for gaming, video conferencing and augmented reality that need rapid responses and a small jitter;
- the *geo-distribution and ubiquity* of fog nodes, more diffused than centralized cloud servers, but less diffused than unintelligent sensors;
- the *awareness of location and context*, useful for those applications where information about the place plays a major role;
- the exploitation of *fast mobility*, such as is the case with smart connected vehicles;
- the *predominance of wireless* on-demand communications and access;
- the attention to *large-scale distributed control systems*, such as smart grids, smart traffic systems, and so forth;
- the importance of *streaming and real-timeliness*, especially for continuous monitoring and early detection of events [33] or delay-sensitive applications, rather than batch elaborations;
- the *heterogeneity* of the nodes composing the fog layer and of the environments they are deployed in;
- the possibilities of *interoperability and federation* across different domains and the chance to seamlessly move computation from one fog node to another one.

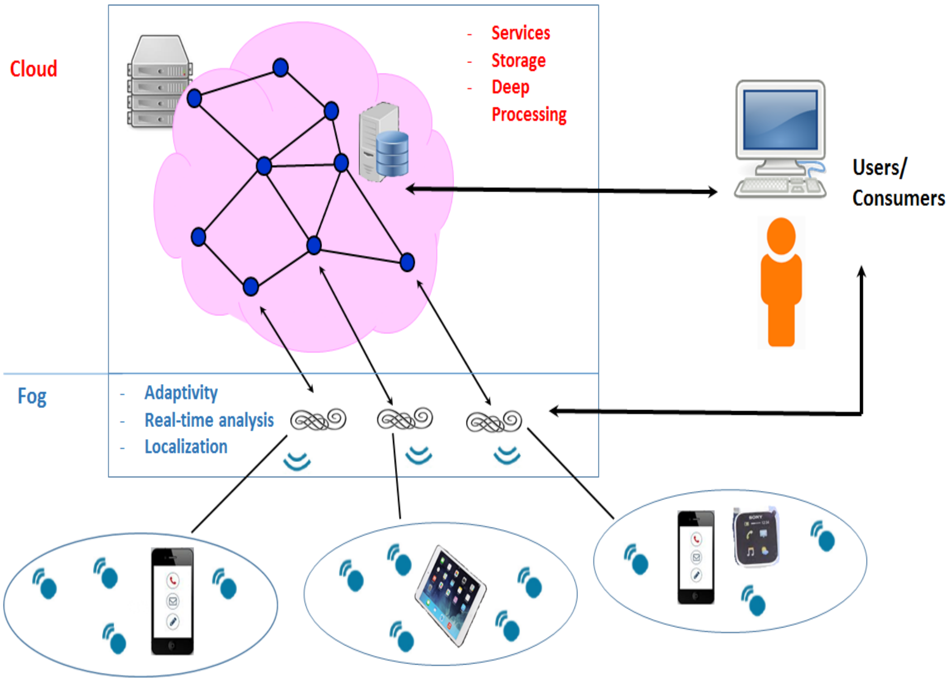


Figure 1. A cloud and fog-based architecture. The LANs are composed by general purpose smart objects (filled circles), smartphones, tablets, wearables, and so on.

Overall, the cloud remains an important player in the global architecture, mainly thanks to its ability to concentrate powerful data centers, as well as to provide a global view of the whole architecture. Moreover, the cloud can also be changed into a consumer of big data streams, coming from the fog layers, in a sort of hierarchy that could encompass more than one fog tier, as well as intra-layer and inter-layer communications. Both the cloud and the fog could be a part of the communication with the final users, providing different, but complementary services in a sort of continuous cooperation. The former could be in charge of historical and large datasets and of deep and long analyses of big data, whereas the latter could focus on shorter response times, localization information and flow control, by enabling fog gateways with computing capabilities. This can foster throughput increase, energy savings and consolidation of the available resources in turn [34]. Moreover, the fog could be employed in the case of disasters or failures of the cloud, as a distributed backup of that information being stored and maintained within the cloud itself. Furthermore, and more importantly, using fog computing reduces the amount of data transferred to the cloud, lessening the amount of needed bandwidth and fostering the security of the data themselves, by refraining from transferring those pieces of information that may result in being risky in terms of privacy and legal regulations.

However, fog nodes present also some drawbacks, compared to the cloud, that should be carefully considered:

- they cannot be regarded as reliable or trusted as cloud servers; furthermore, identifying and repairing a fog node is not trivial;
- they are not able to guarantee lasting network connectivity;
- scheduling tasks are complicated, and there still is not 100% security for where to perform operations such as workload balancing [35], system monitoring, and the like;
- their heterogeneity may lead to sharp differences in resources among nodes and thus hinder cooperation;
- mobility may interrupt running processes on fog nodes, and some other ways to provide it in a seamless way must then be enacted;

- security and privacy are a concern for low-powered and small-resourced nodes that may contain sensitive data and could be easily assailable or that can support different and non-interoperable cryptographic standards;
- data management and service discovery are challenging, as well as is choosing the best algorithms in order to shuffle data among devices;
- data consistency and redundancy may be problematic to manage, even if the final outcome could be better than using the cloud exclusively.

Nevertheless, as some recent outages [36] confirm, the cloud seems no more adequate in keeping up with the ongoing Internet evolution and the need for new architectures based on smart access networks, and fluidity in distributing in-network functions is compelling, even in those scenarios involving distance learning. That is why in the following, we present our proposal for a fog-based VLE architecture, enriched with big data stream capabilities, that tries to overcome the drawbacks of cloud-based learning environments.

4. Big Stream Fog-Based e-Learning Architecture

In this section, we describe in detail the proposed architecture, based both on big data streams and fog computing techniques, which we argue can offer to a distance learning institution the benefits described in Section 6. The framework we devised is based on Apache Storm as a stream-processing engine and on SAMOA as a distributed machine learning framework. Storm was chosen because of its inherent master-slave architecture, which can easily mirror the division between the cloud and the fog, and also because it seemed more suitable to the management of different streams of data, as Samza is more message-oriented and more reliant on external facilities. We could use Spark streaming capabilities; however, they rely on batching of data updates at regular intervals, and this is different from what Storm performs, since Storm handles each event individually, allowing for a task parallelism rather than a data computation parallelism. Moreover, the Spark data parallel paradigm requires a shared file system for optimal use of stable data, while we focused on direct exploitation of streams of data rather than on their distributed storage. Finally, SAMOA was chosen because it integrates perfectly with Storm, but it allows one to handle also different stream processing engines, resulting in an extensible solution.

First of all, in the following subsections, we describe the general architecture, then the integration of Storm in a cloud- and fog-based e-learning scenario; then, we detail the fog layer and finally point out the sources of learning data and how they are mined through machine learning techniques.

4.1. General Architecture

The general scenario we envision is depicted in Figure 2, and it features a constant interplay between the cloud and the fog through inner APIs.

The cloud part is in charge of historical backup and heavy storage operations, as well as mining techniques requiring much time. This portion is to provide long-term forecasts and scores selected features communicating, through outer APIs, with macro-level and intermediate-level users like educational policy makers and educational institution managers.

The fog part is composed of lightweight distributed NoSQL storage facilities, as well as various Storm slaves (local area network gateways or even smart sensors and gadgets), instead. They are used to perform short-term predictions through light mining techniques, in order to calculate various features and to interface, always through the outer APIs, with other kinds of consumers, such as teachers, tutors and students themselves, in order to provide useful suggestions.

The outer API could work according to a subscribe/notify paradigm, allowing some nodes to act as brokers; this implies that external stakeholders subscribe to certain streams of data or predictors, and they receive information whenever it is available through some notifications. On the other hand, the inner API, works according to a push/pull pattern where requests and commands are mainly driven by the cloud layer, while streams of data come mainly, but not only, from the fog layer, as will be clear in the following. Moreover, the inner APIs drive both control and data information, while the outer APIs usually carry only informative content, in order to suggest to different stakeholders how to fine-tune their own learning-related activities.

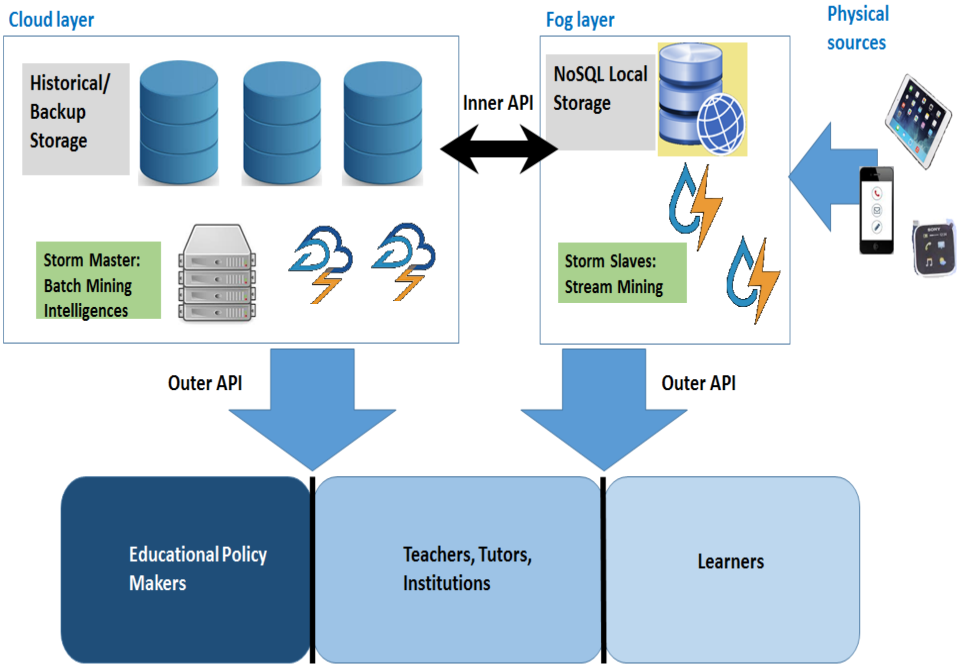

Cloud layer
Historical/ Backup Storage
Storm Master: Batch Mining Intelligences
Inner API
Fog layer
NoSQL Local Storage
Storm Slaves: Stream Mining
Physical sources
Outer API
Outer API
Educational Policy Makers
Teachers, Tutors, Institutions
Learners

Figure 2. A general overview of the big stream and fog-based e-learning structure we propose.

4.2. Integration of Storm with the Proposed Architecture

In this subsection, we detail the integration of Storm with the aforementioned architecture. Storm is a stream-processing engine that is free, open source, fast, scalable and fault-tolerant. It is capable of integration with all known Databases (DBs) and to process and partition streams of data as necessary. In Storm, a stream is an unbounded sequence of tuples, themselves processed in parallel in a distributed fashion. The tuple features an ID and a series of fields that in our case can identify the student, the LO, the time of the day that the LO was accessed, the GPS coordinates of the place that the LO is being used in, the time spent on the LO, the teacher who created it, the course the LO belongs to, and so on. Storm's model is based on a topology, a graph connecting spouts and bolts.

The spouts of Storm, that is the sources of data, can be of two different types: reliable and unreliable, i.e., they respectively do not forget and forget the data to be transmitted in case of failure. In our scenario, the spouts can be located both in the smart gadgets, which can be considered both reliable and unreliable sources according to their inherent nature (a sensor can be unreliable, while a smartphone can be reliable instead), and in fog gateways, in this case usually reliable, being capable of re-transmitting data. As explained in Section 4.5, the cloud layer can be considered filled with data sources as well and, as a consequence, possible spouts.

Bolts are processing units performing operations such as filtering, aggregation, interacting with DBs, and the like. They can perform simple stream transformations or be grouped into multi-bolts in order to execute complex tasks and being sources, in turn, of multiple streams. In our architecture, bolts are located in fog gateways as slayers of more powerful master bolts located in the cloud. Cloud-bolts are in charge of directing the work of distributed fog-bolts across the network by using inner APIs, based on lookup and retrieval procedures as the ones described in Section 4.4. Moreover, by means of the stream grouping feature, cloud- and fog-bolts are also able to decide which streams they receive and how to partition them across different tasks of a certain bolt, e.g., in a shuffle mode, on the basis of certain fields such as the student ID, and so on.

Storm provides an abstraction that allows one not to have to consider how the parallel tasks are performed, i.e., more worker processes are active for the same topology and, inside each worker, various threads execute different tasks for various spouts and bolts. In our architecture, however, worker processes are located only in cloud servers and in fog gateways, rather than in the ultimate smart devices.

4.3. Details of the Fog Layer

In Figure 3, we focus our actions on the fog layer describing more in detail the sub-levels of which it is composed. These layers are the classic ones for the cloud computing infrastructure, but in this case, they are shared across different devices, be they IoT gateways, smart objects themselves, and the like, according to a structure similar to the one described in [37]. The three sub-layers may all coexist in the same equipment or be distributed across different objects. They can be briefly described as follows:

- *Infrastructure as a Service*: providing hardware management concerning computing power, memory and CPU usage, I/O of raw data, monitoring, etc. This layer provides ephemeral storage facilities, as well as monitoring capabilities. We envisaged it to be mostly made up of Storm spouts, i.e., sources of big data streams, collected and aggregated from the available resources. It provides an interface near sensors and actuators. The order of magnitude of its operations is milliseconds.
- *Platform as a Service*: This is a sort of fogviser, similarly to the one presented in [38], able to offer stream management and dynamic resource allocation, as well as semi-permanent storage for a neighborhood or small community of sensors. Being composed of Storm bolts, it also solves possible disputes and furnishes planning and execution capabilities, as well as decision-making opportunities for co-located lightweight analyses or heavy mining performed at the SaaS layer. The order of magnitude of its operations is measured in seconds.
- *Software as a Service*: furnishing big data and big data stream mining functionalities, heavy machine learning capabilities, permanent storage and all supported Business Intelligence models. It is composed of Storm bolts or complex bolt groups, and it provides deep analysis features and extracts valuable information and knowledge from the incoming data streams. This layer is a part only of powerful fog gateways, as well as the cloud. The order of magnitude of its operations is minutes or hours.

In Figure 3, we show only spouts (rectangles) in the IaaS layer, while bolts (circles) and complex bolts are the actual processing units in the PaaS and SaaS layers. Indeed, data sources can be more multifaceted and logically present also in other layers, as will be explained in Section 4.5. Bolts may partially consume data or forward them upwards; moreover, they provide different services, according to the layer they belong to, and they can be, at the same time, listeners of one, or more than one, stream of data (the dashed blue lines) and publishers of new streams, climbing up the layered structure according to different paths. Bolts may feature different complexities as they can either implement basic processing operations or receive processed streams that they elaborate further before delivering to other nodes or the final consumers. As a general rule, to avoid loops, bolts at lower layers cannot be implemented as listeners of nodes of upper levels, i.e., there cannot be streams going from an upper circle to a lower one. This organization is suitable for both minimizing the delay, between the instant of raw data generation and the moment when information is notified to its final consumer, and optimizing resource allocation, as objects with no active bolts can be turned off and temporarily removed from the framework. Moreover, it allows for both hierarchy and parallelism, fostering a reduction in the burdens of communication bandwidth and high computing performances at the same time, through multithreading within the bolts themselves.

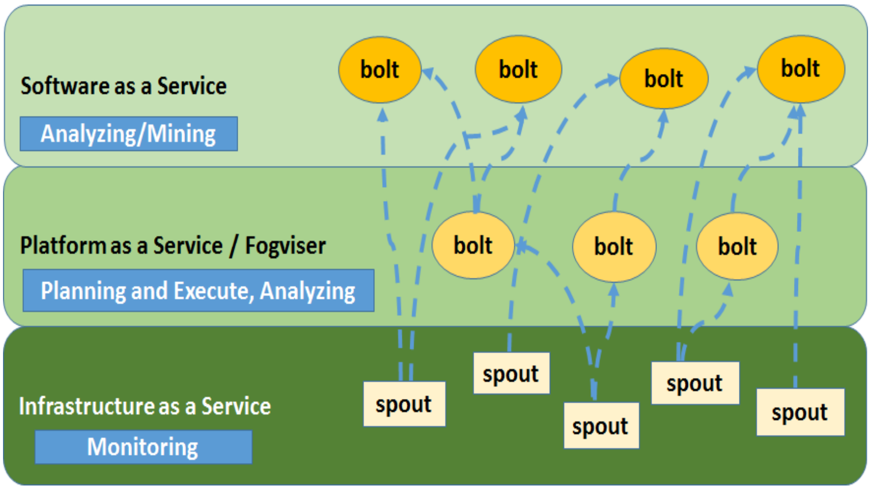


Figure 3. Details of the fog layer where different big data streams (dashed lines), as well as Storm's spouts and bolts are visible.

4.4. Workflow Management and Resource Lookup

Two important aspects in distributed architectures, such as the one presented in this article, regard service and data discovery and retrieval, as well as workflow management. In the following, we detail these aspects with reference to the aforementioned scenario.

Concerning service and data discovery, our proposal could be based on Distributed Hash Tables (DHT), not unlike what usually happens in structured peer-to-peer networks such as Chord or Kademlia [39]. The peers we consider in this case are fog nodes, endowed with at least the PaaS layer and capable of semi-permanent storage and dynamic source allocation and retrieval. In this way, fog gateways may be federated through the DHT overlay network across different domains, furnishing at the same time scalability, quick lookups and a global view as a legacy cloud architecture. In this light, those nodes featuring only spouts register by nodes endowed with upper layer bolts that in turn manage discovery, lookup and access operations according to a certain DHT algorithm, possibly enriched with trust and reputation capabilities [40], without relying on any centralized cloud-based functional entity.

In the scenario we describe in this article, featuring many streams of data and fog layers, one should also carefully consider how to split the workload for those services aggregating information from nearby devices and how to guarantee consistency among shared data. From this perspective, a workflow management layer is shared across the bolts in the SaaS and PaaS layers as described in Figure 3, with the aim of enriching them with synchronization, locking, scheduling and caching capabilities. Caching plays a fundamental role in fine-tuning the transmission rate of data streams from the lower levels towards the same-layer or upper-layer logic components. Locking processes are triggered whenever multiple spouts try to write on the same stream of data, in order to allow for the sources to retain their own data. Scheduling takes place in defining thresholds either for data transmission and migration between different workflow states or for offloading the workload to the cloud layer, even if the process can happen both ways.

4.5. Data Sources

In this subsection, we discuss the main software sources of the data that can be produced and managed by the VLE architecture we envisioned.

The sources described in [4] can be subdivided into the cloud and the fog layers according to Figure 4. The possible categories are the following:

- *Only-Cloud* sources: These include the subcategory of application-general subsystems and modules such as communication services, shared documents, content packaging (SCORM), as well as administrative and business object managers. In this subcategory, one can consider also the recent Learning Record Store (http://tincanapi.com/learning-record-store/) (LRS) modules, able to gather many sources of learning data. Other possible subcategories that contain only-cloud sources are middleware with centralized web servers and administration toolkits.

- *Cloud and Fog* sources: These may be subdivided into the following three subcategories: middleware reusable building blocks, system software and application-specific subsystems:
  - The first include services such as APIs and DBMSsthat could be distributed and differently characterized across the cloud and the fog;
  - The second encompasses operating systems (present also in smart devices) and network protocols (be they the traditional TCP/IP stack for cloud servers or the lightweight 6LowPAN and DTLS for IoT gadgets and gateways);
  - The third revolves around user profile managers (portfolios, working groups, etc.), course management modules, educational resource delivery tools (tracking of results and of time of studying) and student support systems (recommendations, bookmarks, statistics, shared annotation spaces, etc.)
- *Only-Fog* sources: These include physical sources such as laptops, tablets, smartphones, smartbands and smartwatches, as well as their embedded sensors (GPS, gyroscopes, accelerometers, etc.) and the other sensors not embedded and positioned in the IaaS layer of Figure 3.

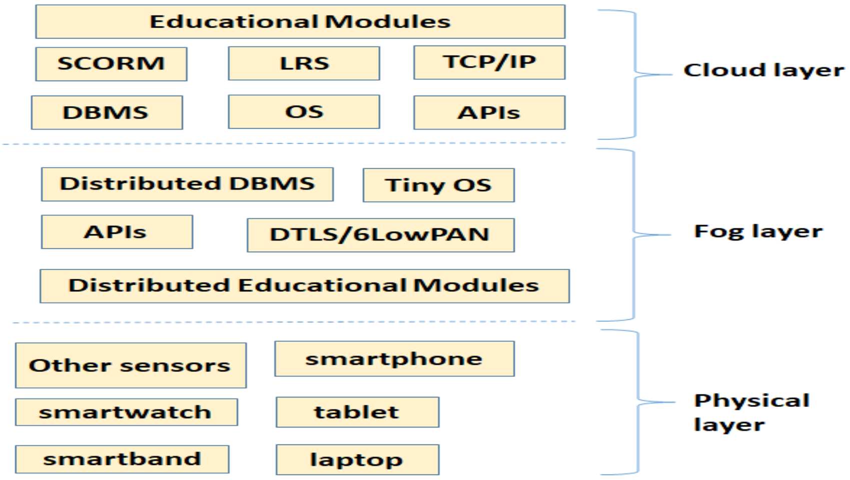


Figure 4. Different big data stream sources in different layers. SCORM, Shareable Content Object Reference Model; LRS, Learning Record Store.

4.6. SAMOA Stream Mining

In this subsection, we describe how machine learning and mining techniques can be usefully employed to extract useful hints, for all educational stakeholders, from the information generated by the big data stream and fog-based VLE architecture we envisioned, and particularly by the aforementioned data sources.

These techniques can enrich the learning model of academic institutions by shortening response time, fostering in-time feedback to teachers, estimating skills, tuning didactics strategies, detecting early drop outs [41], allowing for the analysis and visualization of data, for the creation of feedback supporting instructors, for the recommendations and forecasts of students' performances, to construct courses and tools, as well as to detect students' behavior [4]. For this purpose, we employed SAMOA, and we are going to describe its integration in the architecture we envision by making reference to the structure depicted in Figure 5 that is superimposed over the one in Figure 3 and is more detailed:

sources are present in all layers, according to Figure 4; moreover, we added the cloud level on top of the fog sub-layers.

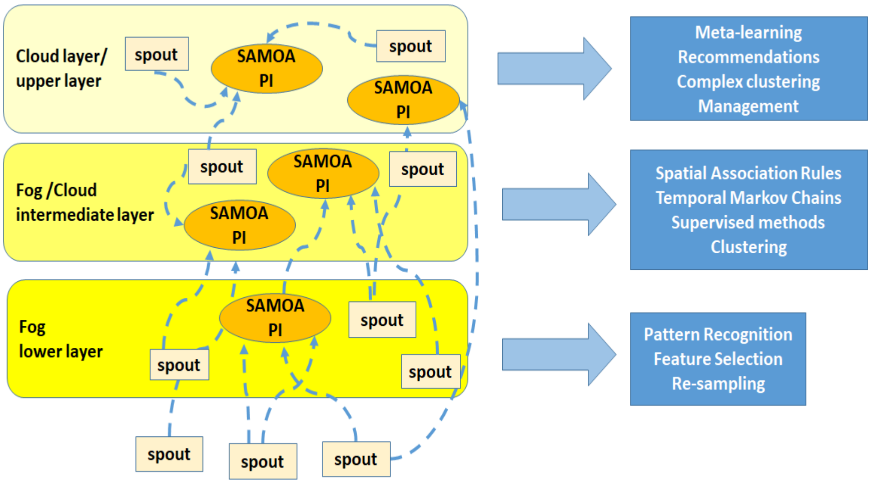


Figure 5. Scalable Advanced Massive Online Analysis (SAMOA) integrated with the envisioned architecture. PI, Processing Item.

SAMOA is a distributed framework allowing its users to apply machine learning algorithms to the huge streams of data managed through a stream processing engine, like the aforementioned Storm. It is based on a series of Processing Items (PIs), or processors, that receive streams of data and perform subsequent analyses. These PIs, in our architecture, may coincide with the bolts described above, as processors are simply containers implementing machine learning algorithms, and they may reside in those fog nodes capable of a minimum level of computation, e.g., fog gateways. This is possible thanks to the ability SAMOA has to reuse the existing computational infrastructure and to perfectly integrate itself with Storm. As a matter of fact, SAMOA is both a framework and a set of libraries, but examined in its entirety, it can be considered as a distributed platform, wherein each processing element provides proper and advanced machine learning and classification algorithms across all system layers.

Some examples of techniques SAMOA can help to implement are supervised learning algorithms for classification such as Support Vector Machines (SVM), k-nearest neighbor algorithms and neural networks. An important technique in this category is the Vertical Hoeffding Tree (VHT), which allows improving vertical computation parallelism across attributes of different tuples. For example, some PIs are in charge of elaborating students' IDs, some others the times of the day for accessing certain LOs, and so on. The PIs keep track of sufficient statistics to analyze the corresponding attributes, thus reducing memory usage and parallelizing the computation of fitness functions for split decisions, similarly to what was done in [42]. Nevertheless, clustering methods are supported as well. An example is CluStream, for clustering time-evolving streams of data. This may take place by building a micro-cluster across the two uppermost sub-levels of the fog layer depicted in Figure 5.

SAMOA may provide further advanced implementations of methods such as bagging and boosting; however, what is important is also its flexibility and adaptability to the architecture we envisioned. In particular, according to Figure 5 and considering the different layers, the machine learning algorithms of SAMOA can be opportunely stratified according to the following layers:

- *The lower layer*, between the IaaS level and the PaaS level in Figure 3, should provide pattern recognition for those data streams arising from sensors and smart objects. Control outputs, as well as particular features, be they selected or extracted, may arise from this stage going directly to the above computing layers, be they in the cloud or in the fog, for further analyses. An important role of this layer is also to tune proper thresholds for the predictive mechanisms of the following layers, e.g., the level of usage of an LO under which a student may be considered at risk of possible withdrawal. Finally, at this level, also opportune techniques for data re-sampling in the presence of unbalanced training datasets can take place. They are very useful for creating a more balanced data class distribution by modifying the training dataset, for example in case the focus of the research is a usually under-represented group, such as the one of students dropping out or the one of people with special needs [43].

- *The intermediate layer*, between the PaaS level and the SaaS level of Figure 3, should be able to provide spatial-temporal association rules, as well as hidden Markov models, suitable for event recognition and for inferring relationships between events and underlying unobserved variables, through the study of occurrence and transition probabilities in a time-evolving manner. An example could be the identification of alerting signals of early dropouts of students. These signals should be analyzed and interpreted quite quickly, as the sooner the problems are identified, the better the retention strategies could work [44]. Therefore, the architecture we envisioned is very useful for the so-called Early Warning Systems (EWS) that, analyzing almost in real-time absenteeism at exams, low performances, limited usage of LOs, and so on, can predict and anticipate dropouts, and thus, effective countermeasures can be enacted by instructors and educational managers. This will be explained in the case study described in Section 5, where supervised classification algorithms will be used in order to identify risky students, and non-supervised machine learning algorithms, such as clustering, could be used to detect in time anomalies and deviations in the behavior of students.
- *The upper layer*, confined in the cloud, is characterized by complex system-wide behavior analyses. An example could be the application of meta-learning techniques, such as the multi-label classification proposed in [45] or the group recommendation strategy explained in [46], to be used in helping instructors to select on-the-fly the best classification algorithms to analyze particular novel and unseen students' data subsets, by means of proper recommendations. This is especially important in the educational scenario, since common users may not be familiar with data mining and machine learning techniques, and the usage of opportune techniques, especially if comprehensible and interpretable, such as decision trees and rule-based algorithms, can lead to better knowledge of the underlying educational phenomena with respect to black-box models. The interplay between the cloud and the fog is, however, necessary since the extraction of specific features from particular datasets is performed with the help of the lower layers of the fog side.

5. Preliminary Case Study

In this section, we present a possible case study of the proposed architecture, demonstrating how it can be used in preventing dropouts.

Teachers and educational managers are going to face both a new group of university students, as well as already enrolled students. The students have different educational needs and learning paths, and teachers would like to minimize dropouts and improve at the same time the usage of the VLE by the students. It is proven that the decision of students to persist or drop out of their studies is related, and quite strongly, to their degree of academic and social integration at university; as a consequence, the big data stream and fog-based architecture we envisioned can be opportunely used to monitor this situation and consequently foster proper retention strategies.

In this specific case study, we refer to eCampus University (http://www.uniecampus.it), one of the Italian distance learning universities, and in particular to the first period of the 2017–2018 academic year: from August 2017–December 2017. The students considered in the study belong to the Faculties of Engineering, Psychology and Literature, and they encompass 100 new enrolled students (40 for courses belonging to Psychology, 40 for Engineering and 20 for Literature), as well as 100 students already enrolled in past years (with the same figures per faculty). Among these 200 students, 100 of them were asked to employ the VLE in a traditional way (Group A), while the other ones (Group B) were asked to employ the new envisioned architecture in its preliminary implementation, which is described in the following.

5.1. Implementation

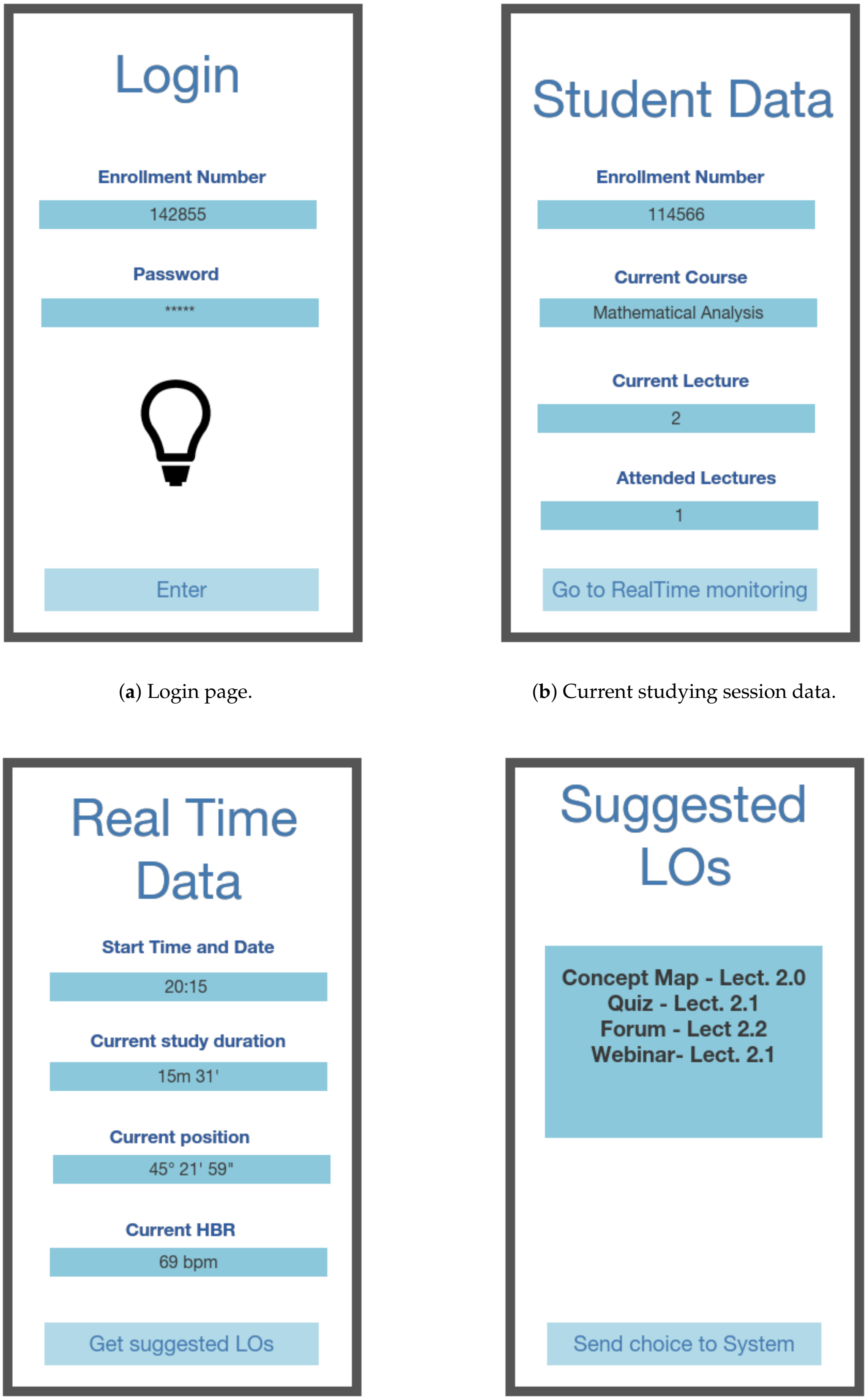
To monitor the students and their studying habits, in this preliminary implementation, we considered simply data coming directly from the smartphones of the students themselves. The students had to own an Android smartphone with at least Version 4.1 (Jelly Bean) of their operating system; moreover, we endowed each student with a proper smartband, i.e., a Xiaomi AMAZFIT Smartband (http://www.huami-usa.com/), and we asked them to wear it when studying. We implemented a basic Android app, whose main screenshots are depicted in Figure 6. This app takes advantage of Storm API Version 1.1.1 and SAMOA API Version 0.4.0. For the sake of simplicity, SAMOA tasks were concentrated in the students' smartphones, while the SAMOA topology was made of PIs interfacing directly with GPS, timing and Heartbeat Rate Variability (HRV) sensors.

The data collected by the implemented app are, as a consequence, the following:

- GPS data coming from the smartphone of the students itself;
- timing data coming from the smartphone of the students itself;
- Heartbeat Rate Variability (HRV) coming from the smartband.

We considered HRV as a possible index of fatigue and drowsiness of the autonomous nervous system activity, as studied in [47]. This datum is retrieved from the app installed on the smartphone and that can communicate via Bluetooth with the smartband. HRV is computed considering a temporal window of 1 min and averaging data sampled at a 1-ms rate. Data are collected by the app when the student starts a studying session inserting the data of the course and of the attending lesson (Figure 6b). This is a voluntary task the students involved in this study were requested to perform before actually studying. As can be seen from Figure 6c, having selected the data of the studying session, the student can look at his/her real-time data: location, time of starting of the studying session and its duration, as well as heart beat rate. On this page, the student can press the button at the bottom to get a list of LOs he/she can choose to study, depicted in Figure 6d. Finally, the student can select one or more LOs, among the ones suggested, and press a button to send feedback on the used LOs to the cloud part of the VLE, as well as to the teachers.

In a future and more complex implementation, some further data could be collected about past education experiences and past social and private backgrounds of the students during their enrollment procedure, through opportune forms and questionnaires. Some examples concern grade point average in secondary school, type of secondary school, family income level, parents' level of education, parent's jobs, if any, presence of certified disabilities in the students or family members, age, gender, and the like. These data could go directly to the cloud level, where the the upper mining layer resides, and could be used for suggesting to teachers and educational managers the most suitable algorithms to be employed for analyzing the incoming group of students. Moreover, the upper layer could listen continuously to data collected from the lower layers in the fog, to enact adaptive changes in the algorithms recommended to instructors and managers, according to the specific situation. However, these data were not used in this simple and preliminary implementation, where a single suggesting algorithm is used, namely vertical Hoeffding tree.



Login

Enrollment Number

142855

Password

*****

Student Data

Enrollment Number

114566

Current Course

Mathematical Analysis

Current Lecture

2

Attended Lectures

1

(**a**) Login page.

(**b**) Current studying session data.

Real Time Data

Start Time and Date

20:15

Current study duration

15m 31'

Current position

45° 21' 59"

Current HBR

69 bpm

Suggested LOs

Concept Map - Lect. 2.0
Quiz - Lect. 2.1
Forum - Lect 2.2
Webinar- Lect. 2.1

(**c**) Real-time data of the student.

(**d**) Suggested Learning Objects.

Figure 6. Main screenshots of the implemented app.

5.2. Data Workflow

In this subsection, we describe in detail the workflow of data in the implemented architecture and the interactions of the users with it.

The data flow diagram of Figure 7 shows which data are collected by the basic implementation of the proposed VLE architecture and how they flow through its various layers. First of all, the lower fog layer collects heart beat rates of the students and their variations and sends them through Bluetooth technology to the smartphone of the students themselves. The smartphones are located in the fog intermediate layer, where SAMOA PIs are located. Here, the implemented app collects HRV data, together with GPS and timing data of the current studying session. GPS data are used by a proper SAMOA PI to find out a possible location through Google Maps services, such as residential districts, restaurants, libraries, etc. These data are used by other SAMOA PIs to determine an opportune list of suggested LOs for the current studying session of the student. This is done according to the vertical Hoeffding tree algorithm. The training dataset is made of different data. Timing data are provided by the university studying habits in the previous three academic years (2016–2017, 2015–2016 and 2014–2015) in the same considered faculties. HRV labeled data were the ones used in [47], while location data were labeled as more or less suited for studying by hand according to three categories: studying place, leisure place, neutral place. Finally, once the student had chosen one of the provided LOs, this information was sent directly to the teachers' smartphone, running an opportune version of the implemented app, and to the cloud part of the VLE. Teachers could verify the chosen LO and, in that case, suggest a different LO to the student via a feedback channel. In the end, data stored in the cloud can be finally used by researchers to perform analyses on the usage of the proposed architecture, as well as on the percentage of dropouts. This is described in the following subsection.

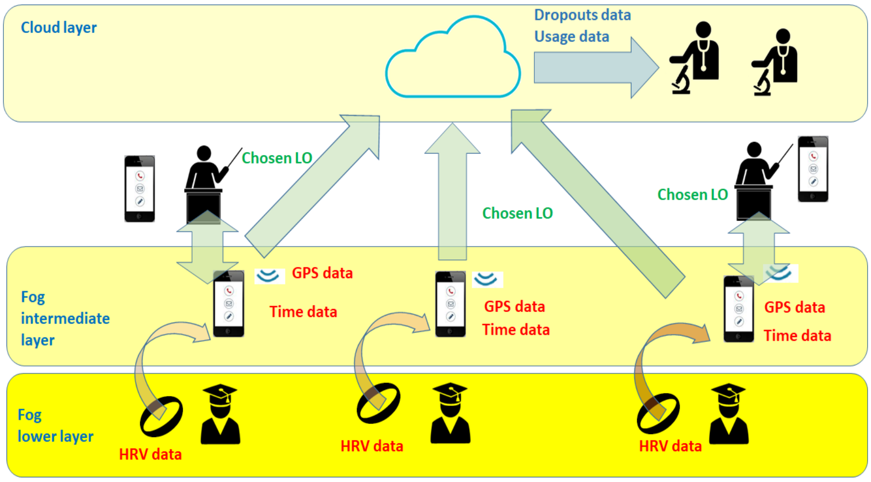


Figure 7. Workflow of data in the basic implementation of the proposed architecture. HRV, Heartbeat Rate Variability.

5.3. Preliminary Results

In this part, we show some interesting preliminary results in the employment of the envisioned architecture in its first basic implementation. The results refer to the period ranging from August 2017 (start of academic year 2017–2018) to December 2017, which encompasses the first two exam weeks for all the considered faculties. The first period of exams took place in September 2017 until the first week of October 2017, while the second period of exams took place from mid-October 2017 until the end of November 2017. Two groups of students were considered: the first one (Group A) encompasses 100 students using the traditional VLE, while the second one (Group B) regards 100 students employing the proposed architecture as described in Section 5.1.

In Figure 8, we show the percentage of dropouts in both considered groups of students versus time, in the first term of academic year 2017–2018. As can be inferred, the percentage of dropouts is considerably limited for Group B (maximum 10%), while it increases till almost 30% for Group A. As for the absolute figures, this means that, in Group B, 10 students dropped out, while, in Group A, 27 students dropped out of university at the end of the considered period. This demonstrates the successfulness of the proposed architecture, at least in this first basic implementation. The graph of Group A undergoes two sharp increases around the second half of September and the first half of November. This could be explained considering that in those periods, the Engineering exams took place and that they are usually considered by the students the most difficult ones, e.g., mathematical analysis. Consequently, dropouts could increase after or during those periods. This does not happen for students using the improved VLE architecture, whose curve undergoes a smooth increase till a maximum of 10%, almost a third compared with the level reached by Group A.

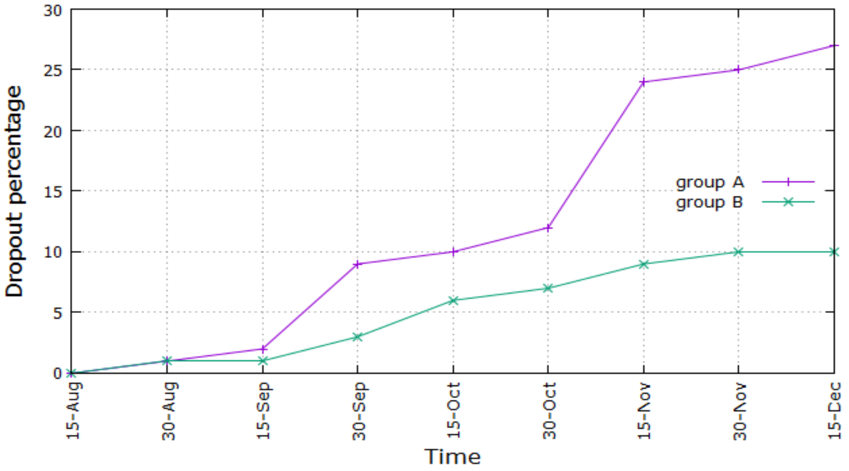


Figure 8. Percentage of dropouts versus time in the first term of academic year 2017–2018 for two groups of students. Group A represents students using traditional VLE, while Group B represents students employing the new proposed architecture.

In Figure 9, we studied the use of the VLE on the part of the students. In this way, we could evaluate whether the students appreciated the new architecture or not. The graph depicts the percentage of usage of the VLE and the improved VLE, by considering students that employed the VLE at least once a day. Particularly, in mid-December, this reached 92 for Group B and 70 for Group A. As one can see, the percentage of usage increases over time in both cases, but students of Group B seem to be more attracted toward the VLE. Moreover, as per the aforementioned absolute figures, the total number of students that employed the VLE at least once a day is greater in the case of Group B than in the case of Group A in mid-December, but this is true also considering previous months. Finally, both graphs show two sharp increases, around the first half of September and the second half of October, during exam sessions.

In Table 1, we detail the figures in dropout reduction for each faculty and for each LO suggested by the implemented architecture. As we can see, even if all suggested solutions are effective at reducing dropouts, their effect could differ across faculties. For example, Engineering students are more influenced by interactive exercises rather than by concept maps, while the opposite takes place for students belonging to the Faculties of Psychology and Literature. Quizzes reach good results especially for Literature students, while, the usage of wikis and forums shows the least effectiveness in reducing dropouts; thus, their usage could not be considered as a proper retention strategy.

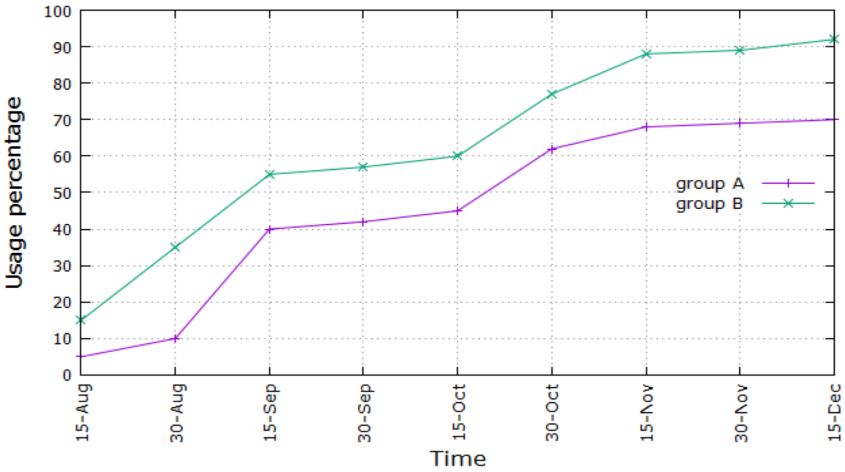


Figure 9. Percentage of usage of the Virtual Learning Environment (VLE) versus time in the first term of academic year 2017–2018 for two groups of students. Group A represents students using traditional VLE, while Group B represents students employing the new proposed architecture. Only students that used the VLE at least once a day are considered.

Table 1. Effectiveness in reducing dropouts of various solutions suggested by the proposed VLE architecture for students of different faculties.

	Engineering	**Psychology**	**Literature**
Interactive exercise	−51%	−32%	−21%
Quizzes	−22%	−34%	−46%
Concept maps	−25%	−60%	−55%
Wikis	−24%	−31%	−22%
Forum	−10%	−5%	−7%

5.4. Future Developments and Usages

The considered implementation is very preliminary and employed to have a proof of concept of the proposed architecture, as well as to obtain some preliminary results validating the described intuitions. As a matter of fact, various envisioned features have not been implemented: fog gateways, different from the smartphones themselves, unreliable Storm bolts located in smart sensors, orchestrating powerful bolts in the cloud, meta-learning capabilities in the cloud, the distributed data discovery and retrieval through a DHT, etc.

In a future implementation, other opportune features such as the levels of drowsiness during the fruition of LOs, skipping of LOs, time spent over certain LOs, number of times an LO is accessed, number of assignments performed, connection bandwidth employed when studying, etc., could be considered. These and the already considered features could be fine-tuned according to (i) the algorithms chosen by the instructors and recommended in the cloud, as well as to (ii) the particular set or subset of students. Finally, the upper layer of the fog tier could perform more complex analyses and elaborations on the selected features, as well as on other characteristics received less frequently and directly from the spouts of the lower layers or even from the cloud. These summarize characteristics such as the average score obtained in the performed assignments, the total time spent on a certain course or a certain assignment, the average mark over different subjects, the time effectively spent in using the libraries, and so on.

The proposed architecture, once entirely implemented, should enable teachers and educational managers to enact prompt retention strategies before students decide to withdraw, as well as tracking who dropped out and why. As a consequence, this information could give feedback to course authors, find relationships between educational patterns and dropouts, discover students' difficulties and common mistakes, guide towards the best fitting transfer model for students with special needs or that need extra help, and the like.

Moreover, important knowledge about why students drop out could be mined. For example, some rules can show that if a student spends a lot of time on the wiki and on the interactive activities, then the probability of dropping out of the university decreases by a certain percentage and with a certain confidence. Conversely, another rule gained from experience could be that if a student with special needs performs less than a certain threshold of quizzes in simultaneously studied subjects, then it is very probable that he/she will fail all exams and consequently quickly drops out, or again, that some features are completely irrelevant to the prediction of withdrawals, or again that the small usage of the forum is not always an index of dropping out if it is joined with frequent access to the LOs, and so on. These deductions and conclusions should be obviously correlated with the features of the considered set of students, i.e., their social provenance and school background, gender, age, and the like, and not simply generalized to the whole population of possible students.

6. Advantages of Big Stream and Fog Computing for e-Learning Environments

One of the central points of this paper is to demonstrate that big data stream mining and fog computing applied to distance learning environments may foster many benefits, both adding new and innovative management features, as well as improving some drawbacks of already presented cloud-based e-learning infrastructures. Therefore, in this section, we discuss the aforementioned advantages, pointing out also some possible issues, by subdividing the pros into three main categories. They are depicted in Figure 10 and can be summarized as follows:

1. those benefits already garnered by cloud-based infrastructures and valid also for big stream fog-based architectures;
2. the disadvantages of cloud-based infrastructures solved partially or wholly by passing to the new proposed solutions;
3. new improvements, ensured only by applying big stream and fog computing to e-learning platforms.

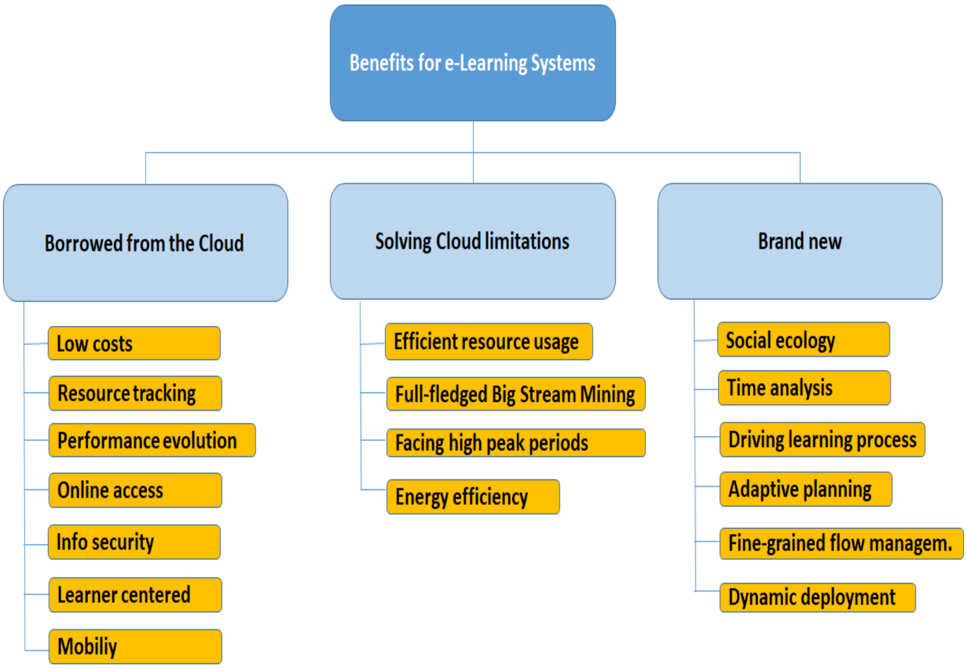


Figure 10. Big stream and fog computing may benefit e-learning infrastructures according to three different types of advantages.

6.1. Benefits Borrowed from Cloud-Based Frameworks

In this category, we can classify features already present in cloud solutions that apply also using a fog-based e-learning platform, maybe with some improvements. They are the following:

- *low costs*, to avoid hardware and software belonging to the learning institution itself. This obviously implies the usage, on the part of the students, of their own devices, be they wearables, tablets or smartphones;
- *tracking* of resources' configuration and utilization even more improved and granular;
- *promotion* of the evolution or extinction of e-learning contents and services, as well as performance improvement;
- *online access* to educational services both for teachers, tutors and students; there is still a complete transparency of the offered applications on the part of the final user;
- *improved information security*; a thief cannot know where the data are stored, either in the cloud or in the fog; moreover, the fog allows for a small amount of data to be transferred to the cloud and thus less bandwidth and encryption requirements. A further advantage may regard granular access control, which different from a coarse-grained cloud-approach, may guarantee sounder security of educational data and foster the students' willingness to share a paramount quantity of data without compromising secrecy;
- a learner-centered approach, by considering the learner as the consumer named in [37];
- even *more mobility* allowed to the students.

6.2. Advantages Solving Drawbacks of Cloud Solutions

This second group is very limited and usually involves the following:

- *efficient resource usage*, as only the interested local devices and fog gateways may be deployed;
- the *full-fledged application of big data stream mining* solutions for deploying on-the-fly machine learning techniques; this is straightforward and comes from the definition of big data stream mining provided in the previous sections of this article;
- the *differentiated flow management* may better face high peak periods followed by idle times: heavy usage is unlikely to happen in a big stream fog-based architecture, since it is learner-centered or, should it happen, at least in a very reduced way.
- *energy efficiency*, as lightweight low-power devices are employed.

6.3. Novel Benefits

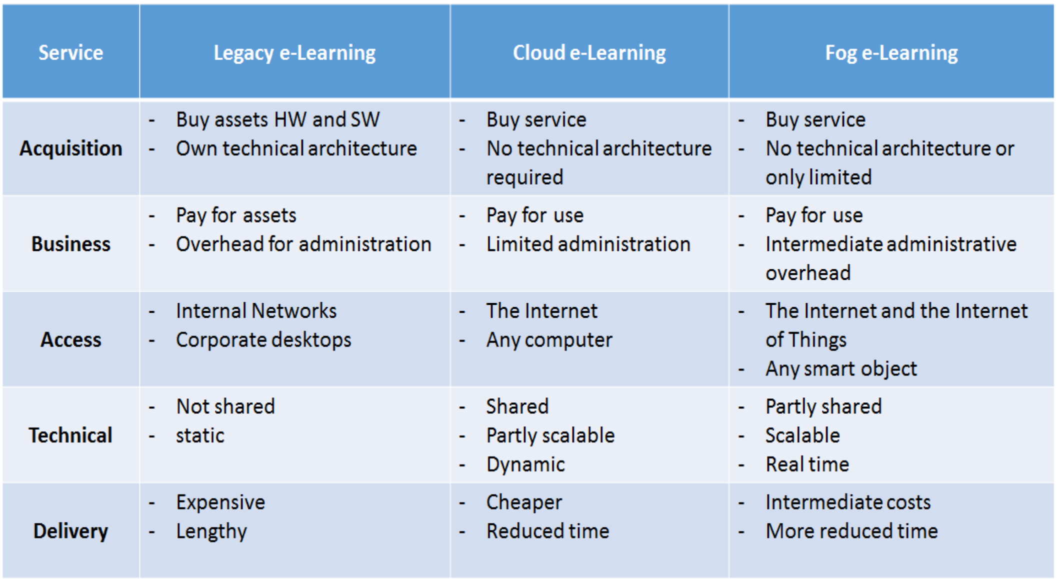
The third type of advantages may be claimed as the most interesting. This encompasses:

- a *social ecology localization* [48] of the students during the learning process: at home, at a restaurant, walking, at the hospital, at school, etc: Thus, teachers can monitor which parts of their subject are mainly learned in particular places and adapt their didactic strategy according to the location. On the other hand, students may tune the lesson (audio, video, slide, tests, etc.) according to the specific place;
- a *temporal analysis* of the students' habits: Performing a granular analysis closer to the students, data regarding the time of day when they make use of various learning instruments may be directly sent to the interested teachers or educational managers, with the ultimate aim of fine-tuning the didactic materials. Moreover, through opportune mining strategies, a studying pattern of different student profiles may be drawn, e.g., working students may prefer to study overnight or during weekends;
- an *accurate driving of the learning process* of the students, providing him/her with real-time feedback from either the instructors or the tutors: Moreover, thanks to the constant real-time monitoring of the students' progresses and results, institutions may provide a personalized student model, therefore customizing learning activities for single students or for groups;

- an *adaptive planning* and scheduling of learning activities, as well as on-the-fly courseware construction;
- a *fine-grained data flow management*: Big data stream technology allows managing and addressing data streams coming from particular students directly to the interested teachers or tutors, who in turn, may provide specific and customized suggestions and explanations placing the consumer of e-learning services, the student, at the center;
- a *dynamic and adaptive deployment* of distance learning courses: Through tailored sensors and smart objects [49], fatigue and mental weariness or, on the contrary, an almost perfect clarity of mind may be promptly detected. As a consequence, a dynamic and adaptive learning strategy when distributing the LOs can be enacted: more interactive parts, made for example by tests, sets of questions, use cases, wikis and mental maps, may interchange themselves with more intensive and complex sections.

6.4. Issues and Comparisons

However, some issues are still present since the conceived e-learning architecture takes place as a compromise between traditional distance learning environments and cloud-based ones. Some of the aforementioned points are highlighted in Figure 11, together with some benefits, according to different service models, coming from [50], for analyzing and subdividing an educational distance learning environment. The models are five, namely: the acquisition one, regarding the purchase of hardware and software assets, the business one, concerning the payments and administrative overhead, one focusing on access, implying the methodologies of admittance to the educational service, the technical one, related to technical features such as dynamicity, scalability and the like, and finally, the one involving delivery, with respect to the costs, as well as the latency of the distribution of e-learning materials.



Service	Legacy e-Learning	Cloud e-Learning	Fog e-Learning
Acquisition	- Buy assets HW and SW - Own technical architecture	- Buy service - No technical architecture required	- Buy service - No technical architecture or only limited
Business	- Pay for assets - Overhead for administration	- Pay for use - Limited administration	- Pay for use - Intermediate administrative overhead
Access	- Internal Networks - Corporate desktops	- The Internet - Any computer	- The Internet and the Internet of Things - Any smart object
Technical	- Not shared - static	- Shared - Partly scalable - Dynamic	- Partly shared - Scalable - Real time
Delivery	- Expensive - Lengthy	- Cheaper - Reduced time	- Intermediate costs - More reduced time

Figure 11. Comparison of traditional e-learning systems, cloud-based ones and fog-based ones according to different service models.

The main issues that are still unresolved, which can be extrapolated from the table, are:

- the need for a certain degree of maintenance, proving to be slightly greater than that of cloud solutions, but still smaller than that usually connected with traditional proprietary e-learning systems;
- the greater number of update operations required, even if more simple and lightweight and always transparent to the end user, as they cannot be instantaneous;

- the uncertain degree of savings in the hardware and software costs, in case student's devices are not usable or compatible;
- a certain degree of overhead for data portability in case local smart nodes or network collectors fail;
- the security issues related to battery-powered devices [51].

7. Conclusions and Discussion

In this article, we have proposed a big data stream and fog-based e-learning framework in order to put together various benefits in favor of distance learning institutions, instructors and students. We provided the state of the art of current VLEs, according to the different stakeholders of an educational scenario, pointing out some of their limitations, and then, we introduced big data stream mining and fog computing as technologies capable of partly solving the explained issues of current solutions. The envisioned framework has been then thoroughly described, taking advantage of a Storm master-slave structure and of a traditional layered architecture enriched with big data stream mining capabilities arising from SAMOA. Moreover, a preliminary implementation of the proposed architecture has been described together with some results demonstrating its feasibility and usefulness.

The advantages that our proposed architecture may foster are multifaceted: real-time correct prediction of students' performances and fatigue, discovery of new didactic models according to location and times of studying, enhancement of students', or groups of students', learning abilities or their engagement in some particular topics.

However, these will be possible provided other issues are addressed in advance such as: technologies integration, user acceptability and innovation in fog gateways, sensors, gadgets and smart objects in general. Some further theoretical issues, to be investigated in the future, regard also the security perspective, such as ownership and privacy of data.

Future research directions may concern both theoretical and practical aspects. In the first case, we can consider the theoretical study of differentiated behaviors between groups and single individuals, as well as the accurate analysis of hidden information, as the quality of the insights is not always guaranteed, especially when crowdsourcing, based on sensing devices owned by the students themselves, is employed. In the second case, we can list a further development of the proposed proof of concept implementation, through a full-fledged development of smart gateways, or the usage of artificial-intelligence capable smartphones, endowed with full Storm and SAMOA elements, the implementation of the distributed DHT and workflow management considering also the whole capabilities of the cloud and different predictive algorithms to compare. Moreover, further metrics and analyses could be performed such as a study of the marks and performances of the students and a finer degree of personalization of the recommended LOs, exploiting long-term data in the cloud. Moreover, evaluating response times and data transmission rates towards the cloud and across different fog layers, as well as the exploitation of social relationships among students and their own devices are further research directions that could empower and enrich the presented architecture.

Acknowledgments: The author acknowledges Prof. Pietro Ducange and Prof. Massimo Vecchio for their valuable help in suggesting ideas for this paper, and Mr. Antonio Enrico Buonocore for the careful proofreading of the paper.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Ducange, P.; Pecori, R.; Mezzina, P. A glimpse on big data analytics in the framework of marketing strategies. *Soft Comput.* **2017**, 1–18, doi:10.1007/s00500-017-2536-4.
2. Belli, L.; Cirani, S.; Davoli, L.; Ferrari, G.; Melegari, L.; Picone, M. Applying Security to a big stream cloud Architecture for the Internet of Things. *Int. J. Distrib. Syst. Technol.* **2016**, *7*, 37–58.
3. Bonomi, F.; Milito, R.; Natarajan, P.; Zhu, J. Fog computing: A Platform for Internet of Things and Analytics. In *Big Data and Internet of Things: A Roadmap for Smart Environments*; Springer: Cham, Switzerland, 2014; pp. 169–186.

4. Ducange, P.; Pecori, R.; Sarti, L.; Vecchio, M. Educational big data Mining: How to Enhance virtual learning environments. In Proceedings of the International Joint Conference SOCO'16-CISIS'16-ICEUTE'16, San Sebastián, Spain, 19–21 October 2016; Springer: Cham, Switzerland, 2017; pp. 681–690.
5. González-Martínez, J.A.; Bote-Lorenzo, M.L.; Gómez-Sánchez, E.; Cano-Parra, R. Cloud computing and education: A state-of-the-art survey. *Comput. Educ.* **2015**, *80*, 132–151.
6. Schneider, B.; Blikstein, P. Unraveling Students' Interaction Around a Tangible Interface using Multimodal Learning Analytics. *J. Educ. Data Min.* **2015**, *7*, 89–116.
7. Clement, B.; Roy, D.; Oudeyer, P.Y.; Lopes, M. Multi-Armed Bandits for Intelligent Tutoring Systems. *J. Educ. Data Min.* **2015**, *7*, 20–48.
8. Stricker, D.; Weibel, D.; Wissmath, B. Efficient learning using a virtual learning environment in a university class. *Comput. Educ.* **2011**, *56*, 495–504.
9. Mueller, D.; Strohmeier, S. Design characteristics of virtual learning environments: State of research. *Comput. Educ.* **2011**, *57*, 2505–2516.
10. Auvinen, T.; Hakulinen, L.; Malmi, L. Increasing Students Awareness of Their Behavior in Online Learning Environments with Visualizations and Achievement Badges. *IEEE Trans. Learn. Technol.* **2015**, *8*, 261–273.
11. Strobel, J.; Wang, J.; Weber, N.; Dyehouse, M. The role of authenticity in design-based learning environments: The case of engineering education. *Comput. Educ.* **2013**, *64*, 143–152.
12. Valenzuela-Valdés, J.F.; Pardo, P.J.; Padilla, P.; Lozano-Guerrero, A.J. Low Cost Ubiquitous Context-Aware Wireless Communications Laboratory for Undergraduate Students. *IEEE Trans. Learn. Technol.* **2016**, *9*, 31–36.
13. Hung, P.; Lam, J.; Wong, C.; Chan, T. A Study on Using Learning Management System with Mobile App. In Proceedings of the International Symposium on Educational Technology (ISET), Wuhan, China, 27–29 July 2015; pp. 168–172.
14. Gilman, E.; Milara, I.S.; Cortés, M.; Riekki, J. Towards User Support in Ubiquitous Learning Systems. *IEEE Trans. Learn. Technol.* **2015**, *8*, 55–68.
15. Limongelli, C.; Lombardi, M.; Marani, A.; Sciarrone, F.; Temperini, M. A recommendation module to help teachers build courses through the Moodle Learning Management System. *New Rev. Hypermedia Multimed.* **2016**, *22*, 58–82.
16. Sergis, S.; Sampson, D.G. Learning Object Recommendations for Teachers Based On Elicited ICT Competence Profiles. *IEEE Trans. Learn. Technol.* **2016**, *9*, 67–80.
17. Hew, T.S.; Kadir, S.L.S.A. Understanding cloud-based VLE from the SDT and CET perspectives: Development and validation of a measurement instrument. *Comput. Educ.* **2016**, *101*, 132–149.
18. Chao, K.M.; James, A.E.; Nanos, A.G.; Chen, J.H.; Stan, S.D.; Muntean, I.; Figliolini, G.; Rea, P.; Bouzgarrou, C.B.; Vitliemov, P.; et al. Cloud E-learning for Mechatronics: CLEM. *Future Gener. Comput. Syst.* **2015**, *48*, 46–59.
19. Gómez, J.E.; Huete, J.F.; Hernandez, V.L. A Contextualized System for Supporting Active Learning. *IEEE Trans. Learn. Technol.* **2016**, *9*, 196–202.
20. Caminero, A.C.; Ros, S.; Hernández, R.; Robles-Gómez, A.; Tobarra, L.; Granjo, P.J.T. VirTUal remoTe labORatories Management System (TUTORES): Using cloud Computing to Acquire University Practical Skills. *IEEE Trans. Learn. Technol.* **2016**, *9*, 133–145.
21. Al-Hudhud, G. Aspect oriented design for team learning management system. *Comput. Hum. Behav.* **2015**, *51*, 627–631.
22. Masud, M.A.H.; Huang, X. An E-learning System Architecture based on cloud computing. *Int. J. Comput. Electr. Autom. Control Inf. Eng.* **2012**, *6*, 255–259.
23. Han, I.; Shin, W.S. The use of a mobile learning management system and academic achievement of online students. *Comput. Educ.* **2016**, *102*, 79–89.
24. Fernández, A.; Peralta, D.; Benítez, J.M.; Herrera, F. E-learning and Educational Data Mining in cloud computing: An Overview. *Int. J. Learn. Technol.* **2014**, *9*, 25–52.
25. Bifet, A. Mining big data in Real Time. *Informatica* **2013**, *37*, 15–20.
26. Erl, T.; Khattak, W.; Buhler, P. *Big Data Fundamentals: Concepts, Drivers & Techniques*; Prentice Hall Press: Englewood Cliffs, NJ, USA, 2016.
27. Leavitt, N. Storage Challenge: Where Will All That big data Go? *Computer* **2013**, *46*, 22–25.

28. Carbone, P.; Gévay, G.E.; Hermann, G.; Katsifodimos, A.; Soto, J.; Markl, V.; Haridi, S. Large-Scale Data Stream Processing Systems. In *Handbook of Big Data Technologies*; Springer: Cham, Switzerland, 2017; pp. 219–260.
29. Ramírez-Gallego, S.; Krawczyk, B.; García, S.; Woźniak, M.; Herrera, F. A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing* **2017**, *239*, 39–57.
30. Tennant, M.; Stahl, F.; Rana, O.; Gomes, J.B. Scalable real-time classification of data streams with concept drift. *Future Gener. Comput. Syst.* **2017**, *75*, 187–199.
31. Dastjerdi, A.V.; Gupta, H.; Calheiros, R.N.; Ghosh, S.K.; Buyya, R. Fog computing: Principles, Architectures, and Applications. *arXiv* **2016**, arXiv:1601.02752.
32. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and Its Role in the Internet of Things. In Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 17 August 2012; ACM: New York, NY, USA, 2012; pp. 13–16.
33. Azimi, I.; Anzanpour, A.; Rahmani, A.M.; Liljeberg, P.; Salakoski, T. Medical warning system based on Internet of Things using fog computing. In Proceedings of the 2016 International Workshop on Big Data and Information Security (IWBIS), Jakarta, Indonesia, 18–19 October 2016; pp. 19–24.
34. Hao, Z.; Novak, E.; Yi, S.; Li, Q. Challenges and Software Architecture for fog computing. *IEEE Int. Comput.* **2017**, *21*, 44–53.
35. Cimorelli, F.; Priscoli, F.D.; Pietrabissa, A.; Celsi, L.R.; Suraci, V.; Zuccaro, L. A distributed load balancing algorithm for the control plane in software defined networking. In Proceedings of the 24th Mediterranean Conference on Control and Automation (MED), Athens, Greece, 21–24 June 2016; pp. 1033–1040.
36. Solon, O. Glitch in Amazon Web Servers Causes Problems for Popular Sites. 2017. Available online: http://www.theguardian.com/technology/2017/feb/28/amazon-web-server-crash-internet-problems (accessed on 3 January 2017).
37. Belli, L.; Cirani, S.; Ferrari, G.; Melegari, L.; Picone, M. A Graph-Based cloud Architecture for big stream Real-Time Applications in the Internet of Things. In Proceedings of the Advances in Service-Oriented and Cloud Computing: Workshops of ESOCC 2014, Manchester, UK, 2–4 September 2014; Revised Selected Papers; Springer International Publishing: Cham, Switzerland, 2015; pp. 91–105.
38. Albeanu, G.; Popentiu-Vladicescu, F. A reliable e-learning architecture based on fog-computing and smart devices. In Proceedings of the 10th International Scientific Conference on eLearning and Software for Education, Bucharest, Romania, 24–25 April 2014.
39. Pecori, R.; Veltri, L. Trust-based routing for Kademlia in a sybil scenario. In Proceedings of the 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 17–19 September 2014; pp. 279–283.
40. Pecori, R. S-Kademlia: A trust and reputation method to mitigate a Sybil attack in Kademlia. *Comput. Netw.* **2016**, *94*, 205–218.
41. Knowles, J.E. Of Needles and Haystacks: Building an Accurate Statewide Dropout Early Warning System in Wisconsin. *J. Educ. Data Min.* **2015**, *7*, 18–67.
42. Sani, L.; Amoretti, M.; Vicari, E.; Mordonini, M.; Pecori, R.; Roli, A.; Villani, M.; Cagnoni, S.; Serra, R. Efficient Search of Relevant Structures in Complex Systems. In Proceedings of the AI*IA 2016 Advances in Artificial Intelligence: 15th International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, 29 November–1 December 2016; Springer: Cham, Switzerland, 2016; pp. 35–48.
43. Luna, J.M.; Romero, C.; Romero, J.R.; Ventura, S. An evolutionary algorithm for the discovery of rare class association rules in learning management systems. *Appl. Intell.* **2015**, *42*, 501–513.
44. Márquez-Vera, C.; Cano, A.; Romero, C.; Noaman, A.Y.M.; Mousa Fardoun, H.; Ventura, S. Early dropout prediction using data mining: A case study with high school students. *Expert Syst.* **2016**, *33*, 107–124.
45. Olmo, J.L.; Romero, C.; Gibaja, E.; Ventura, S. Improving Meta-learning for Algorithm Selection by Using Multi-label Classification: A Case of Study with Educational Data Sets. *Int. J. Comput. Intell. Syst.* **2015**, *8*, 1144–1164.
46. Zapata, A.; Menéndez, V.H.; Prieto, M.E.; Romero, C. Evaluation and selection of group recommendation strategies for collaborative searching of learning objects. *Int. J. Hum. Comput. Stud.* **2015**, *76*, 22–39.
47. Vicente, J.; Laguna, P.; Bartra, A.; Bailon, R. Drowsiness detection using heart rate variability. *Med. Biol. Eng. Comput.* **2016**, *54*, 927–937.

48. Xu, F.; Li, Y.; Chen, M.; Chen, S. Mobile cellular big data: Linking cyberspace and the physical world with social ecology. *IEEE Netw.* **2016**, *30*, 6–12.
49. Akram, S.; Javaid, N.; Ahmad, A.; Khan, Z.A.; Imran, M.; Guizani, M.; Hayat, A.; Ilahi, M. A Fatigue Measuring Protocol for Wireless Body Area Sensor Networks. *J. Med. Syst.* **2015**, *39*, 1–15.
50. Bahrami, M.; Riahi, G. E-learning Systems Based on cloud Computing: A Review. *Proc. Comput. Sci.* **2015**, *62*, 352–359.
51. Merlo, A.; Migliardi, M.; Caviglione, L. A survey on energy-aware security mechanisms. *Perv. Mob. Comput.* **2015**, *24*, 77–90.