Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at *Information* **2019**, *10*, 34; doi:10.3390/info10020034

Article

HOLMeS: eHealth in the Big Data and Deep Learning Era

Flora Amato [1], Stefano Marrone [1], Vincenzo Moscato [1], Gabriele Piantadosi [1], Antonio Picariello [1], Carlo Sansone [1]

[1] Department of Electrical Engineering and Information Technology (DIETI), University of Naples Federico II, via Claudio 21, 80125, Naples, Italy.; name.surname@unina.it

* Correspondence: carlo.sansone@unina.it; Tel.: +39-081-7683-640

Version November 15, 2018 submitted to Preprints

Abstract: Data collection and analysis are becoming more and more important in a variety of application domains as long as the novel technologies advance. At the same time, we are experiencing a growing need for human-machine interaction with expert systems pushing research through new knowledge representation models and interaction paradigms. In particular, in the last years *eHealth* - that indicates all the health-care practices supported by electronic elaboration and remote communications - calls for the availability of smart environment and big computational resources. The aim of this paper is to introduce the *HOLMeS (Health On-Line Medical Suggestions)* framework. The introduced system proposes to change the eHealth paradigm where a trained machine learning algorithm, deployed on a cluster-computing environment, provides medical suggestion via both chat-bot and web-app modules. The chat-bot, based on deep learning approaches, is able to overcome the limitation of biased interaction between users and software, exhibiting a human-like behavior. Results demonstrate the effectiveness of the machine learning algorithms showing 74.65% of Area Under ROC Curve (AUC) when first-level features are used to assess the occurrence of different prevention pathways. When disease-specific features are added, HOLMeS shows 86.78% of AUC achieving a more specific prevention pathway evaluation.

Keywords: eHealth; big data; deep learning; watson; spark; decision support system; prevention pathways

---

1. Introduction

Data collection and analysis are becoming more and more important in a variety of application domains as long as the novel technologies advance. At the same time, we are experiencing a growing need for human-machine interaction with expert systems, pushing research trough new knowledge representation models and interaction paradigms [1].

This can be observed in many fields, from commercial to medical diagnostics. For example, the world-famous retail business company Wal-Mart Stores Inc. produces more than one million customer transactions per hour, representing the record of every single purchase by their point-of-sale terminals in each of their 6000 stores worldwide. A traditional data warehouse able to contain such amount of information should be sized more than 3 Petabytes. This implies that suitable data models and cluster-computing framework are mandatory in order to use machine learning algorithms tailored to this kind and amount of data. As well known, such approaches aim to extract patterns indicating the effectiveness of the business strategies, improving the inventory and supply chain management [2].

Similar problems could arise in medical image processing applications that require elaborating huge amount of data burdened from strong temporal constraints, computationally heavy tasks and privacy issues. For example, a common-size clinical centre equipped with Magnetic Resonance Imaging (MRI) appliances, can provide up to 20-30 MRI scans per day producing about 6 Gigabytes of raw data. This amount can easily tenfold when machine learning and pattern recognition are applied,

© 2018 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at Information 2019, 10, 34; doi:10.3390/info10020034

growing up to 30GB per day requiring a suitable storage system, a dedicated computing architecture and specific store-and-retrieve procedures [3,4].

As for medical imaging, patient records should be treated with the same carefulness. A clinical centre may want to store all its patient records within a digital database in order to have historical information, thus strongly improving further diagnosis. A reliable digital knowledge-base should include all the details about personal data (age, height and weight), anamnesis (family history, patient lifestyle and personal health status), clinical investigations (examination results and diagnostic images) [5], the applied treatments and the resulting diagnosis [6]. It is worth noticing that it is very important to store both positive and negative responses in order to have a reliable and complete knowledge-base who referring to. Moreover, each single case may have a specific data structure that may strongly differ from that of other ones. It follows that traditional relational databases may not be able to properly handle such **variety** of data. Therefore, unstructured data storage mechanisms should be used to avoid data boundaries or constraints and guarantee modularity and upgradability of the system.

As preliminary study we analyzed the data-grow models, showing how clinical data can easily increase in terms of both **velocity** and **volume**.

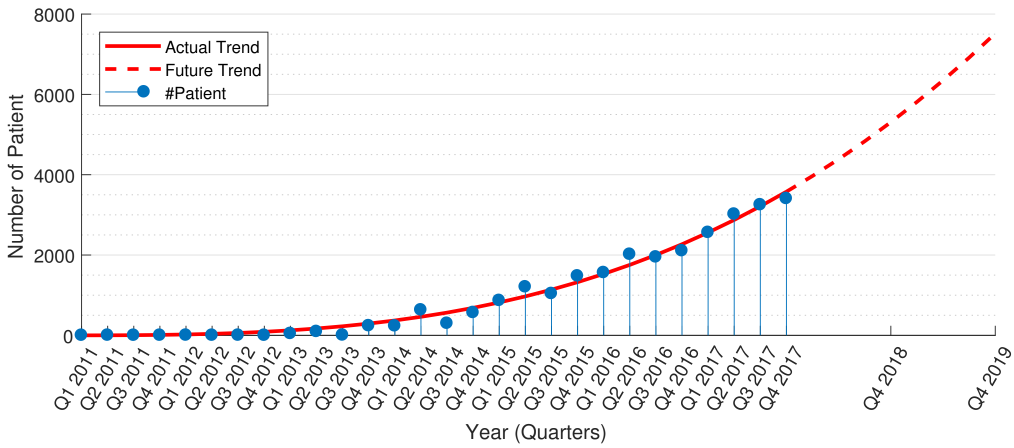

Figure 1. Data production in a medium-size medical centre (see Section 4.4). Growing collection rate of patient data is pretty evident. Tendency curve (in red) shows estimation for future years.

The Figure 1 shows real data collected from a medical examination centre, used in this work to validate the proposed system (see Section 4.4). During the observed period from 2011 to 2017, the collection rate of clinical data shows a rapidly increasing rate. At the end of the observed period, the clinical has a production of about 2000 patient records per quarter. An easy estimation of the growing factor indicates a production of about 5200 records per quarter at the end of the 2018 and of 7500 records in the late 2019. These numbers give an indication of how the data grow quickly promising to reach big volume data in few years. These three features: variety in the data structure, growing velocity of data production and big volumes of data represent the three 'V' of the Big Data paradigm [7] as shown in Figure 2.

The medical context includes up to 10,000 known human diseases, however, a physician is able to recall only a small fraction of these diseases at the diagnosis moment. Even if the medical diagnosis are driven by clinical trials and patient anamnesis, diseases misclassification in the diagnosis phase is frequent. According to a study performed in 2012 in an Intensive Care Unit (ICU) of the USA as many as 40,500 patients die annually due to misdiagnosis [8].

In addition, some additional problems must be taken into account:

- **Data sensitivity** of medical records (privacy must be guaranteed) [3,4]
- **Operational time** comparable with clinical environment time [9,10].
- **Patient trustiness**, meaning that the system should represent a clinician. Therefore, human-like interaction models may be provided with the aim of minimising any kind of bias.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at Information 2019, 10, 34; doi:10.3390/info10020034

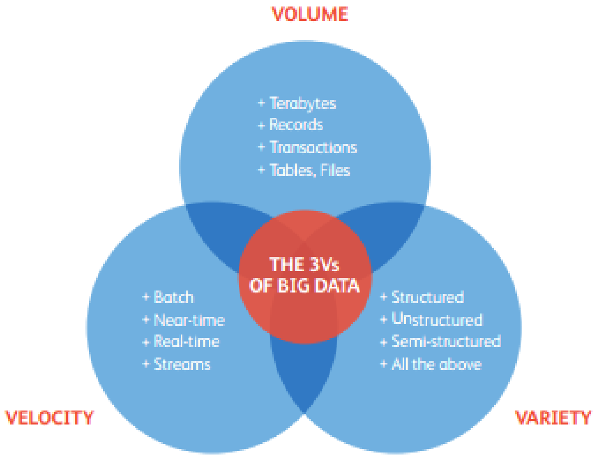

Figure 2. The 3 Vs of Big Data – Volume, Variety and Velocity

- **Massive data handling**, since diagnostic by images produces huge amounts of data (not only structured as 3D volumes) [11].
- **Context scalability**, in order to take into account some sort of distributed computing [3,4].

Those requirements determine to study and realize systems/infrastructure/architecture able to operate on big data analysis, managing at same time computational complexity, scalability, upgradeability and costs [1].

The described need of a smart environment and big computational resources thus push to search proper solutions in the eHealth paradigm. The term *eHealth* (also written e-health) indicates all the health-care practice supported by electronic elaboration and remote communications [12]. Several different definitions of eHealth has been proposed [13]: some authors consider the term as an extension of health informatics or digital processing of health data [14]; another point of view consider under the eHealth paradigm all the health-care procedures delivered via the Internet [15,16]; the term can be also be referred to health-care services delivered via mobile devices (mHealth) [17] and, finally, in a more general meaning, they can also be considered under the eHealth paradigm all the services or systems laying on the edge of health-care and information technology.

In the social network era, communication over the Internet has a strong influence from chat-like conversation model. This new communication approach influenced the social interactions, the business services and the customer care philosophy. health-care and eHealth should benefit from this new communication paradigm performing an human-like interaction that can remove any biasing due to the interaction with an informative system, giving the patient the feeling of a natural conversation and exploiting the possibility of a schema-free conversation interaction. Artificial intelligence, nowadays even more reliable thanks to deep learning approaches, provides automatic and adaptive human-like conversation behaviour via chat-bot applications, enabling user-system interaction able to simulate human behaviour. A chat-bot (also known as a talkbot, chatterbot, Bot, chatterbox, Artificial Conversational Entity) is a software that holds a conversation via message exchange. The final goal of such programs is to convincingly simulate human behaviours to pass the Turing test. Today, chat-bots are used in dialogue systems for several practical purposes including customer service or data collection. The most performing bots use sophisticated Natural Language Processing (NLP) systems and are trained with a deep neural network to better emulate the human behaviours. Many simpler systems scan for keywords within the input, then pull a reply with the most matching keywords, or the most similar wording pattern, from a database. Moreover, many application service providers

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at Information 2019, 10, 34; doi:10.3390/info10020034

(ASPs) deploys chat-bot services to be trained on a specific task via example databases and able to interact with the most spread social network and instant messaging application.

The aim of this paper is to propose a novel eHealth interaction paradigm where a trained machine learning algorithm, deployed on a cluster-computing framework, provides medical suggestion via a chat-bot module. The chat-bot, trained with deep learning approaches, is able to overcome the limitation of biased interaction between the user and the software providing human behaviour. The whole system is called HOLMeS: Health On-Line Medical Suggestions.

The cluster-computing facility is provided by Databricks[1], where a cluster of servers allows computing over a Spark framework. The chat-bot application is provided by the Watson Conversation Service, designed and trained via the Bluemix platform.

The paper is organised as follows: in Section 2 we briefly describe the evolution of Decision Support Systems for eHealth applications, while in Section 3 we show the tools used to design, to implement and to provide the HOLMeS service. In Section 4, we present the proposed system, explaining each module and providing an overview of the complete architecture. Finally, the obtained results are presented in Section 5 and discussed in Section 6, where we also draw some conclusions.

2. State of Art

The eHealth paradigm is not a new approach in supporting physician in the hard task of giving the right diagnosis. For many years, in this context, the Clinical decision support systems[18] (CDSSs) had a fundamental role in assisting physicians and other health professionals with decision-making tasks, such as determining diagnosis from the patient data. The early CDSSs were designed as two-stage interaction system where the physician put the patient data, the symptoms and few outcomes from clinical tests, receiving the diagnosis. The more advanced CDSSs are able to interact with the physicians and guide them in a progressive process of successive refinement till to the final diagnosis.

There are two main types of CDSS:

- The **Knowledge-Based CDSS** consist of three parts: a mechanism to communicate (GUI), that allow the system to show the results to the user as well as have to input into the system; a knowledge-base (KB), that contains the rules and associations of compiled data; the inference engine (IE) combines the rules from the knowledge-base with the patient's data.
- The **The NonKnowledge-Based CDSS**, based on machine learning system which allows learning from past experiences (Ground Truth) and/or finds patterns in clinical data; Two types of non-knowledge-based systems are artificial neural networks (as for developing Computer Aided Diagnosis Systems [19,20]) and genetic algorithms.

The first CDSS (in early 1975) was MYCIN [21], an expert system that operated using a fairly simple inference engine and a knowledge base of about 600 rules to identify bacteria causing severe infections, such as bacteremia and meningitis, and to recommend antibiotics, with the dosage adjusted for patient's body weight. At the same time (in the years 1970-1986) was developed CADUCEUS that worked using an inference engine similar to MYCIN's; it made a number of changes (like incorporating abductive reasoning) to deal with problems such as the additional complexity of internal diseases, a number of simultaneous diseases and generally flawed and scarce data. With DXplain (1984-1986) the CDSS systems began to be web-oriented providing access through the World Wide Web and, in the 90s (precisely with RODIA in 1997), they began to be used in medical imaging, diagnostics, orthopaedic and other more complex medical disciplines. RODIA provides two major functionality areas: image (x-ray and ultrasonography) quantitative evaluation and fractures healing monitoring [22].

Modern CDSS, such as DiagnosisPro [23], developed in the 2000s is also a web-oriented medical expert system that provides exhaustive diagnostic possibilities for 11,000 diseases and 30,000 findings

---

[1] https://databricks.com/

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at Information 2019, 10, 34; doi:10.3390/info10020034

providing the most appropriate differential diagnosis. The actual web-oriented trend is still growing, both in order to facilitate the diffusion and to exploit greater computing capabilities of servers not always available on conventional workstations. Moreover, this allowed to greatly increase the number of diseases and to engage more complex tasks such as the analysis of biomedical images: It has been claimed that decision support will begin to replace clinicians in common tasks in the future [24].

Among the major services nowadays provided under the eHealth paradigm there are:

- Electronic health record: enabling the communication of patient data between different health-care professionals (GPs, specialists etc.);
- Computerized physician order entry: a means of requesting diagnostic tests and treatments electronically and receiving the results
- ePrescribing: access to prescribing options, printing prescriptions to patients and sometimes electronic transmission of prescriptions from doctors to pharmacists
- Clinical decision support system: providing information electronically about protocols and standards for health-care professionals to use in diagnosing and treating patients
- Telemedicine: physical and psychological diagnosis and treatments at a distance, including telemonitoring of patients functions;
- Consumer health informatics: use of electronic resources on medical topics by healthy individuals or patients;
- Health knowledge management: e.g. in an overview of latest medical journals, best practice guidelines or epidemiological tracking (examples include physician resources such as Medscape and MDLinx);
- Virtual health-care teams: consisting of health-care professionals who collaborate and share information on patients through digital equipment (for transmural care);
- mHealth or m-Health: includes the use of mobile devices in collecting aggregate and patient level health data, providing health-care information to practitioners, researchers, and patients, real-time monitoring of patient vitals, and direct provision of care (via mobile telemedicine);
- Medical research using grids: powerful computing and data management capabilities to handle large amounts of heterogeneous data [25].
- Health informatics/health-care information systems: also often refer to software solutions for appointment scheduling, patient data management, work schedule management and other administrative tasks surrounding health

To the best of our knowledge, our proposal, HOLMeS: Health On-Line Medical Suggestions is a novelty in the eHealth field. In fact, it offers most of the services required by the eHealth paradigm in an innovative way.

3. Big Data tools supporting eHealth applications

In recent years the growing amount of data and the need of extrapolating useful information from them, motivated many big players to develop their own deep learning and big data application frameworks. Most common service-oriented architecture are deployed using different service models such as: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [26]. In order to understand the techniques, methods and approaches proposed in this work, it is necessary to describe some of such framework.

3.1. Apache Spark Cluster

Apache Spark is the general-purpose system for cluster computing developed by the Apache foundation [27]. It is designed to provides its services trough high-level APIs available for Java, Scala, Python and R, also supporting different higher-level tools among which it is worth to mention SQL support, Structured Data Processing and MLlib (a machine learning library) [28]. Spark is able to run both by itself, or over three different cluster managers:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at Information 2019, 10, 34; doi:10.3390/info10020034

- **Standalone**, using the included simple cluster manager.
- **Apache Mesos** [29], a more general cluster manager able to run Hadoop MapReduce and service applications.
- **Hadoop YARN** [30], the standard resource manager for Hadoop 2.

3.1.1. Hadoop HDSF

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware [31]. It inherits some characteristics from the others distributed file system, but, by design, it is high fault-tolerant and well suited to work on commodity hardware. It also provides high throughput access to application that use large dataset, even if it is not fully compliant to the POSIX standard in order to provide data streaming. HDFS was originally intended to meet some specific goals:

- **Hardware Failure**, since it has to run over hundreds or thousands of commodity server machines.
- **Streaming Data Access**, since application that use HDFS usually need batch processing and thus hight throughput is preferred over low access latency.
- **Large Data**, supporting ten of millions of files each of size in gigabytes to terabytes.
- **Simple Coherency Model**, in order to simplify application access policies, since many applications usually need a write-once-read-many access model for files.
- **Portability** both across heterogeneous hardware and software, in order to sustain its usage and diffusion.

3.1.2. MapReduce

MapReduce is a software framework originally introduced by Google to support computing on big data set with parallel, distributed algorithm on a cluster [32]. A typical MapReduce application is made by two steps:

- Map step, that filters and sorts data (for example sorting patient by age, arranging a different queue for each possible age value).
- Reduce step, that summarizes the data (for example, counting the number of patients in each queue, producing the age frequencies).

The core application manages all operations, primarily by distributing data over different servers, executing all tasks in parallel, managing communication and fault tolerance schemas. One of the most popular open-source implementation of MapReduce is Apache Hadoop [33].

3.1.3. Spark.ML and ML Pipelines

Spark.ml, introduced starting from Spark 1.2, is a package that collects the inheritance of the old Spark MLlib, standardizing the Spark API for machine learning application. ML Pipelines is a set of high-level API designed to provide a uniform development strategy to help users creating or combining multiple machine learning algorithms into a single Pipeline (also called work-flow) [34]. A Pipeline is a set of subsequent stages, each of which can be either a Transformer (an abstraction including both feature transformers and trained models) or an Estimator (an abstraction of any king of learning concept or algorithm that trains on data). Stages within a pipeline are executed in order, where the output of a given stage represents the input for the subsequent one. Apache Spark ML support several high-level programming languages, including java, python and scala.

3.2. Databricks

Databricks refers both to the company founded by the creators of Apache Spark and to the relative cluster infrastructure [35]. It aims to assist users in developing cloud-based big data processing application using Spark by selling both a hosted cloud product (built on Spark) and relative training

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at Information 2019, 10, 34; doi:10.3390/info10020034

and courses. Databricks provides, in a IaaS model, a virtual analytic platform based on a web-based platform for working with Spark, that provides automated cluster management and IPython-style notebooks.

3.3. Watson

Watson is an advanced question answering infrastructure, developed by IBM, able to interact in natural language processing. It was developed, in 2011, to answer questions on the quiz show 'Jeopardy!' [36] winning the final prize. To achieve such task, the Watson inference engine was able to process structured and unstructured contents producing and elaborating up to four TeraBytes of data. In 2013, IBM devotes Watson to commercial applications involving the supercomputer in a decision support task for lung cancer treatment at Memorial Sloan Kettering Cancer Center, in New York City. From that application, several winning strategy leads IBM in successful customer services application.

In health-care, the natural language skills, the inference engine and the evidence-based learning capabilities are been applied to contribute to clinical decision support systems for use by several medical professionals. Moreover, Watson Cognitive services are able to draw from 600,000 medical evidence reports, 1.5 million patient records and clinical trials, and two million pages of text from medical journals to help doctors develop treatment plans tailored to patients' individual symptoms, genetics, and histories.

To provide advanced services deployed over the Watson supercomputer, IBM releases Bluemix, a cloud platform as a service (PaaS) supporting several programming languages and cognitive services. Among the newer services provided via the Bluemix platform there is Watson Conversation service. IBM Watson Conversation service, can create, via API interfaces, an application that understands natural-language inputs and uses machine learning to respond to customers in a way that simulates a conversation between humans.

4. Methods

In this work we introduce HOLMeS: Health On-Line Medical Suggestions. It is composed of different modules that collaborate each other to provide an advanced eHealth service through an intuitive chat application (Figure 3).

The HOLMeS system general architecture is composed as follows:

HOLMeS Application is the HOLMeS system core. Written in Python, it implements the main logic and orchestrates modules communications and functions. In particular, it communicates with the user trough the chat-bot, interpreting his requests using the functionalities provided by the Watson Conversation API, managing requests and responses between the patient and application modules in order to provide the disease prevention results.

HOLMeS Chat-Bot is the module dedicated to interact with the user, in order to let him feel more conformable. The bot is designed to understand different kinds of chat interactions, from formal writing to more handy ones. It is one of the HOLMeS System entry points and interacts with the user to let him chose the required service. It is also intended to kindly ask the user for required information (such as age, height, weight, smoking status, and so on) just as a human physician would behave.

HOLMeS Web-App is the interface dedicated to performing the medical interview via dynamics web forms. The forms change the question fields according to the previous answers, minimizing the interaction while maximizing the quality of the information. The results are provided by bar plots and gauges. Although the web-form interfaces are not as comfortable as a chat-bot, they offer a suitable modality for internet browsers and mobile devices (via mobile-ready interfaces).

IBM Watson provides the service needed to establish a written conversation, simulating human interactions, trough its Conversation APIs. Main features include natural language processing and text mining trough deep learning approaches.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at Information 2019, 10, 34; doi:10.3390/info10020034

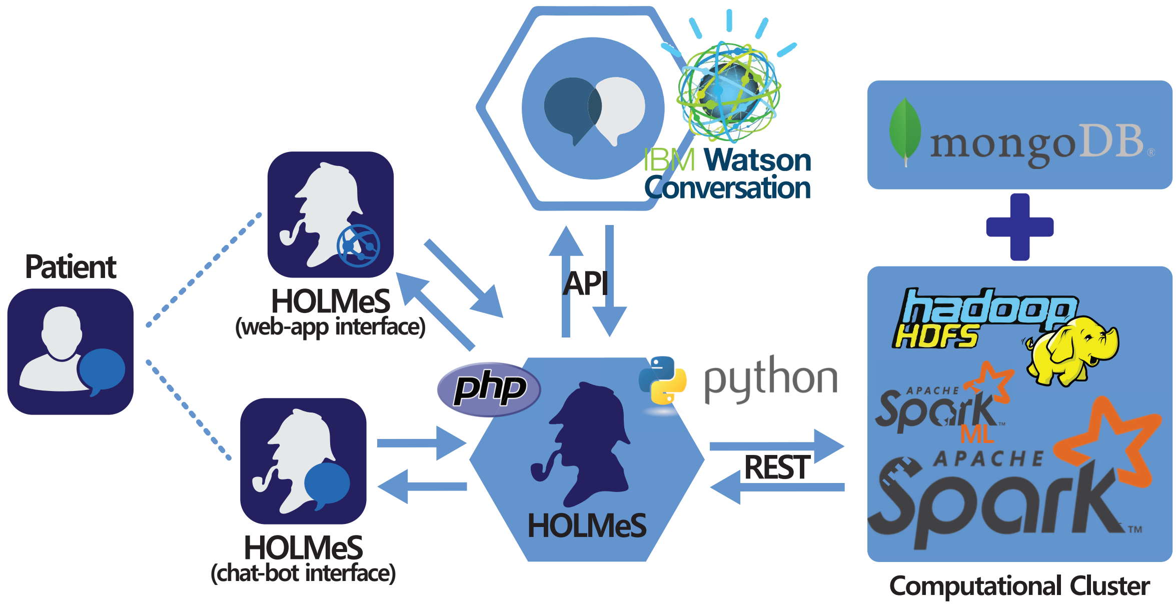

Figure 3. HOLMeS System main modules with interaction paradigm: On bottom-centre the HOLMeS Application core; On the left the HOLMeS Chat-Bot (bottom) and the patient (top) interacting with HOLMeS; On top-centre the IBM Watson Conversation logic adopted by HOLMeS trough its API; On the right the Computational Cluster providing storage, computing and machine learning services used by HOLMeS.

Computational Cluster Implements the decision making logic. It uses the Apache Spark cluster executed over the Databricks infrastructures, in order to be enough fast and scalable to be effectively used in a very big clinical scenario, where many requests from different patients come together, ensuring response time comparable to that of a human physician. It uses machine learning algorithms from Spark ML library, previously trained on many clinical features, in order to predict the expected occurrence probability for different diseases. Finally, the storage service is delegated to Hadoop HDFS and to Mongo DB for storing patient clinical records.

HOLMeS is able to handle four possible use-case scenarios:

1. Provide general information about itself or the affiliated medical centre.
2. Collecting general patient information in order to provide general prevention pathways indications among different disease.
3. Collecting detailed patient information (clinical, examination results and so on) in order to evaluate the probability of needing a specific disease prevention partway.
4. Book a *de-visu* examination with the affiliated medical centre.

As described, HOLMeS was designed to be modular, where each concern is associated with a given module. This allows designing and developing each piece separately, in order to better fit specific requirements such as scalability, speed, availability, and so on. In the following we will describe the HOLMeS main functionalities. Finally, the used dataset will be introduced, laying the foundation for results showed in the section 5.

4.1. The Chat-Bot Interaction Skills

HOLMeS Chat-Bot module mimics humans behaviour in order to let him feel more conformable, overcoming the biasing of a 'machine interaction'. It relies on the IBM Watson Conversation APIs that is able to hold a complete automatic chat with the user. Applying natural language processing and machine learning algorithms, the Watson service identifies the user intents and the concerning entities.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at *Information* **2019**, *10*, 34; doi:10.3390/info10020034

Watson provides high-level dialogue flow design allowing to write down the entire conversation example-by-example. These examples and the interaction model, composed of intents and entities, is used to fine-tune a pre-trained deep neural network.

In order to fulfill the four possible uses cases (described above) Watson conversation service has been designed to recognize the following intents:

1. #greetings: to handle the initial conversation preamble.
2. #book: to describe actions as reserving a *de-visu* examination.
3. #get: to ask for receive something such as prevention pathways indications or information about the centre.
4. #put: to catch the intent of giving the required information to the system.

Moreover, several entities useful to contextualize the above intents has been described. The entities are formalized by synonyms. More equivalent formulations of an entity are provided, the more precise will be the subject recognition. Some of the used entities are in the following list:

1. @HOLMeS
2. @HOLMeS_functionalities
3. @clinical_centre
4. @address
5. @de_visu_examination
6. @specific_patway_examination
7. @general_patways_examination
8. @age
9. @sex
10. @birthday
11. @height

Intents and entities are then combined to achieve a fully automated conversation flow by using the ‘dialog design toolbox’ of the Bluemix platform. For example, to achieve the initial greeting preamble the application can catch the *#greeting* intent and, thus, will answer with random greeting sentences. Only after the greeting step is passed, the user can ask for more intents such as *#get* referring to a specific entity, for example, the *@HOLMeS_functionalities*. In this case, HOLMeS will answer with the list of all the available functionality the user can ask for. An example of the functionalities described above is depicted in the Figure 4.

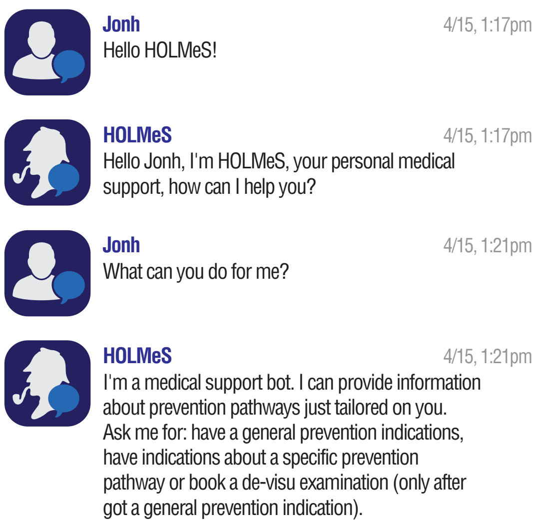

Figure 4. Conversing with HOLMeS: Greating preamble.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at Information 2019, 10, 34; doi:10.3390/info10020034

4.2. The Machine Learning algorithm

Even if able to understand natural language, the chat-bot alone is not able to provide any kind of medical advice. The simpler way to add this feature is to define a static database and let the bot match user clinical information with those stored in the database, returning all the results obtained by this query. This approach has many limitations, among which it is worth to mention:

- A reduced set of possible matches, since the bot can only query the database, without to be able to make inferences on provided data.
- A huge work is required by system designers and medical domain experts that have to code all desired diseases causes, aspects and user clinical matching.
- A very reduced upgradeability, since if a designer wants the bot to be able to recognize a new disease, he has to re-code all the knowledge-base.

The machine learning core aims to overcome all these limitations, since *a)* it is able to derive inferences based on uses clinical data, *b)* autonomously learn the best representation of diseases after a proper training stage and thus *c)* automatically generates the knowledge-base if new diseases have to be introduced to the system. In particular, to enhance HOLMeS upgradeability, we propose to implement a different classifier for each disease the system has to be able to predict. Finally, in order to combine all classifiers results and to return simple and clear information to the user, a suitable combiner is required.

The main advantages of the multi-classifier architecture are:

Improved scalability, since classifiers can be easily distributed over a cluster to accommodate growing amount of user request;

Reduced computational time, since the schema is high parallelizable because different classifier predictions can be evaluated in parallel;

High manutenibility, since any single classifier can be adapted, corrected or improved without any impact on any other system components;

High upgradability, since a new classifier can be easily added to make HOLMeS able to deals with any new desired diseases;

To deploy our machine learning algorithms, we chose to use Spark as cluster-computing framework deployed over the Databricks infrastructures. Spark provides data storage, implicit data parallelism and fault-tolerance features. The Spark.ML library provides all the data-preparation functionalities and the machine learning algorithms to train the Random Forest models using the collected training data. The same Spark.ML library is, then, used to achieve the final classification for each diseases prevention pathway. Moreover, via the 'ML pipeline' paradigm, Spark allows to easily deploy and generate the knowledge-base (represented by the trained model) every time a re-training is required (such as when new diseases or new patients are stored in the database).

To be more specific, the HOLMeS machine learning core provides two different working modalities:

General-level prevention evaluation, in which the user can query the HOLMeS System, trough the chat bot, in order to have first medical prevention advice submitting some simple clinical features such as age, height, weight, living place, smoking status, some diseases familiarity, and so on.

Specific disease prevention evaluation, in which a user can query the HOLMeS System, trough the chat bot, in order to have a more detailed prevention advice about a specific disease by submitting more specific features such as results examination, blood pressure, respiratory and heart rate, oxygen saturation, body temperature, and so on.

The output of the machine learning algorithm, deployed in spark, is different according to the required working modality. When a general-level evaluation is required, the algorithm will produce a histogram graph containing the disease occurrence probability per each of the disease available in the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at Information 2019, 10, 34; doi:10.3390/info10020034

dataset. This result will be stored in the database and can be retrieved by the physician, when a *de-visu* examination will occur or can be used by the patient to chose the specific disease to be evaluated in the second working modality. In this last working modality, the user will receive a probability of be affected by the requested disease.

4.3. General functionalities

The core of the system has in charge to orchestrate the several data flows. Data from and to the patient, through the chat-bot (Fig. 6) or web-app (Fig. 5) interactions, has to be routed to the computational cluster to achieve the final diagnosis. Only the chat-bot skills require the Watson services using Conversation API. The interaction between the user and the application occurs without any further elaboration by the HOLMeS Application that observes the data flow memorizing the information of interest. When interaction requires elaborating machine learning algorithms or accessing to the data storage, the core system contacts the cluster for accomplishing the specific task.

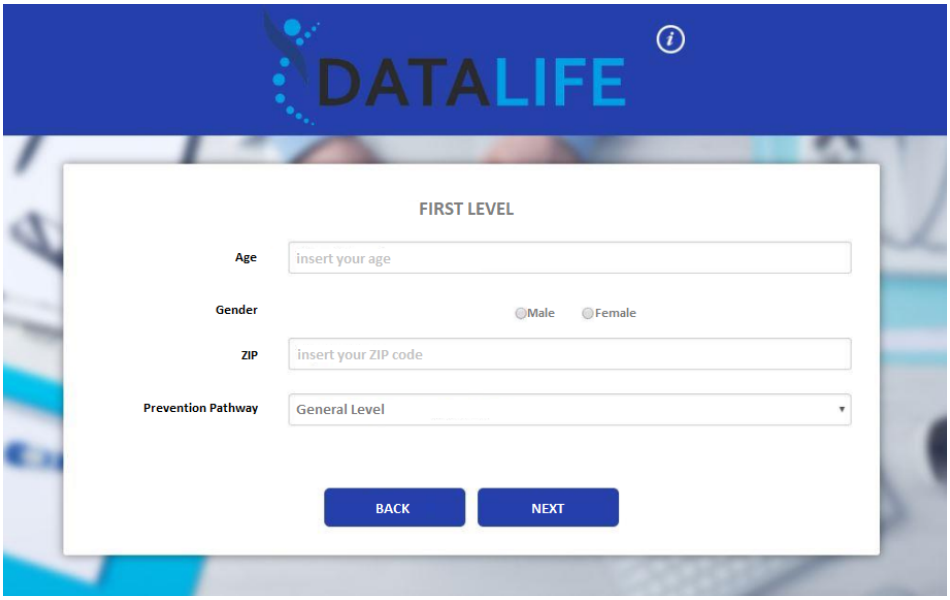

Figure 5. Layout of the deployed web application: interface asking for the preliminary in formations.

In order to fulfill the four possible uses cases (as described above) core application needs to catch the following intents:

- The user wants information about the affiliated medical centre.
- The user asks for general information about the system and its functionalities.
- The patient desires to obtain general-level evaluation about the available prevention pathways.
- The patient desires to obtain detailed indications about a specific disease prevention pathway (only after a general survey has been carried out).
- The patient wants a *de-visu* examination (only after a general survey has been carried out).

For example, to meet the 2^{nd} use-case scenario, after the greeting preamble, HOLMeS recognize the *#get* user intent combined with the *@general_patways_examination* entity. Such interaction yield to the data collecting chat flows with the aim of achieving a general-level prevention pathway evaluation.

The result of a general-level evaluation conversation should be similar to the graph in Figure 7.

4.4. Dataset

In this paper we focus on the preventive health-care for 13 different diseases. A total of 16733 patients prevention records has been collected. Each patient contains a positive or negative ground-truth indicating whether its prevention pathway leads to a positive diagnosis. For each disease, table 1 reports the number of involved patients and their average age.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at *Information* **2019**, *10*, 34; doi:10.3390/info10020034

Figure 6. Conversing with HOLMeS via the chat-bot interface: Asking for general indications about the available prevention pathways.

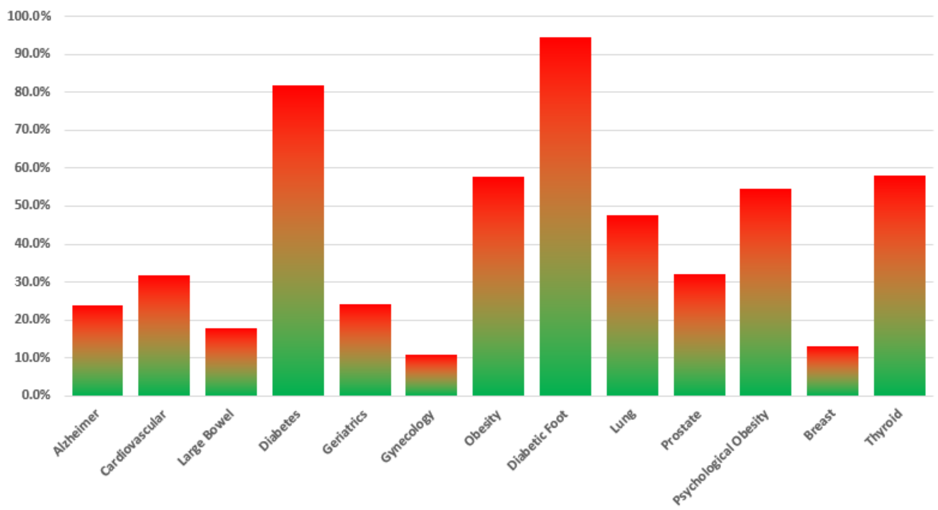

Figure 7. General-level evaluation response (both for chat-bot and web-app interface).

5. Results

In this section we report the results obtained by applying the HOLMeS System to the uses case scenario introduced in the previous section. Given the nature of the problem under consideration, it is crucial to validate the results reliability. For this reason, we performed a 10-fold cross validation for each disease prevention pathway, separately for both working modalities. It is worth noticing that the system has to correctly identify all risky patients, while minimising false alarms. To this aim, we chose to compare classifier results in term of Area Under the ROC Curve (AUC-ROC). For both the working modalities, Table 2 reports the mean values of the 10-fold cross validation evaluated for each disease by means of a Random Forest classifier made of 200 trees and without any limitation on the trees depth; the table also reports the 95% confidence interval and the median value for both working modalities, evaluated over all diseases.

Moreover, it is worth noticing that the chat-bot service can successful held a full conversation with the user (as shown in Figures 4 6), successful providing the final diagnosis.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at Information 2019, 10, 34; doi:10.3390/info10020034

	#Records	Age (AVG)
Alzheimer	176	68
Cardiovascular	299	55
Large Bowel	513	53
Diabetes	3955	60
Geriatrics	89	70
Gynecology	1344	46
Obesity	629	50
Diabetic Foot	1075	65
Lung	225	56
Prostate	1045	60
Psychological Obesity	388	50
Breast	5867	47
Thyroid	1128	46
Total	16733	56

Table 1. Dataset size and stats per each prevention pathway.

	General Pathway (First Level)	Specific Pathway (Second Level)
Alzheimer	67.08% (± 0.06%)	90.60% (± 0.06%)
Cardiovascular	71.35% (± 0.06%)	83.58% (± 0.06%)
Large Bowel	58.07% (± 0.04%)	90.16% (± 0.04%)
Diabetes	78.94% (± 0.03%)	81.21% (± 0.03%)
Geriatrics	69.53% (± 0.09%)	97.79% (± 0.09%)
Gynecology	75.15% (± 0.03%)	79.17% (± 0.03%)
Obesity	81.80% (± 0.04%)	80.20% (± 0.04%)
Diabetic Foot	83.00% (± 0.03%)	90.88% (± 0.03%)
Lung	75.91% (± 0.06%)	96.93% (± 0.06%)
Prostate	83.68% (± 0.03%)	98.82% (± 0.03%)
Psychological Obesity	74.65% (± 0.06%)	80.30% (± 0.06%)
Breast	60.65% (± 0.03%)	62.56% (± 0.03%)
Thyroid	68.27% (± 0.03%)	96.00% (± 0.03%)
Median	74.65%	86.78%

Table 2. Classification performance of the general and specific pathway evaluation (first level) per each disease in the dataset. The performances are in terms of Area Under the ROC Curve (AUC-ROC) obtained from a 10-fold cross validation. 95% confidence intervals and median value are reported.

6. Conclusion

The aim of this paper was to propose HOLMeS (Health On-Line Medical Suggestions). The proposed approach suggests changing the eHealth paradigm by using a trained machine learning algorithm, deployed on a cluster-computing framework, that provides medical suggestion via a chat-bot module. The chat-bot, trained with deep learning approaches, is able to overcome the limitation of biased interaction between the user and the software simulating human behavior.

The results presented in the Section 5 validates the machine learning algorithms both for the general-level prevention indications and for the disease specific prevention evaluation. It is worth noticing that the second-level evaluation leads to better results (improving the AUC of about 12%) because it exploits more specific features provided by physicians or obtained via further examination.

The choice of using a Random Forest classifier was made because when using machine learning in a clinical scenario, it is very important to be able to understand the processes that lead classifier to make its predictions. A Random Forest is an ensemble classifier made of different trees, thus, it is possible to reconstruct the sequence of choices made by the system.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at *Information* **2019**, *10*, 34; doi:10.3390/info10020034

We are currently working on a GUI that allows using knowledge-base and the machine-learning algorithms, deployed in the cluster infrastructure, without conversate with the chat-bot. That will turn HOLMeS into a classical Clinical decision support systems giving to a physician the possibility of continuing the disease prevention evaluation starting from the second-level, when the patient ask for a *de-visu* examination.

Future works will also focus on improving the trustiness of the chat-bot providing more human-like behaviours and improving the so far implemented uses-case scenarios. Moreover, additional prevention pathways should be added and more patient will be recruited in order to have a complete prevention and a more reliable result. Finally, Watson provides support for different natural languages (such as Brazilian Portuguese, English, French, Italian, Spanish, German, Traditional Chinese, Simplified Chinese, Dutch and Arabic). This multi-language feature could be added to our system pushing the spread of HOLMeS in different countries and helping in the medical knowledge sharing.

Author Contributions: All authors read and approved the final manuscript.

Acknowledgments: The authors would like to thank CMO - one of the partner of the DATALIFE research consortium - for the availability of the dataset that allowed us to perform the experimentation of such paper. The authors would also like to thank all anonymous reviewers and editors for their helpful suggestions for the improvement of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Buyya, R.; Yeo, C.S.; Venugopal, S. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on. Ieee, 2008, pp. 5–13.
2. Bryant, R.; Katz, R.H.; Lazowska, E.D. Big-Data Computing: Creating Revolutionary Breakthroughs in Commerce, Science and Society, 2008.
3. Piantadosi, G.; Marrone, S.; Sansone, M.; Sansone, C. A Secure OsiriX Plug-In for Detecting Suspicious Lesions in Breast DCE-MRI. In *Algorithms and Architectures for Parallel Processing*; Springer International Publishing, 2013; pp. 217–224.
4. Piantadosi, G.; Marrone, S.; Sansone, M.; Sansone, C. A secure, scalable and versatile multi-layer client–server architecture for remote intelligent data processing. *Journal of Reliable Intelligent Environments* **2015**, *1*, 173–187.
5. Pérez, A.; Gojenola, K.; Casillas, A.; Oronoz, M.; de Ilarraza, A.D. Computer aided classification of diagnostic terms in spanish. *Expert Systems With Applications* **2015**, *42*, 2949–2958.
6. Arsene, O.; Dumitrache, I.; Mihu, I. Expert system for medicine diagnosis using software agents. *Expert Systems with Applications* **2015**, *42*, 1825–1834.
7. Ongenae, F.; Claeys, M.; Dupont, T.; Kerckhove, W.; Verhoeve, P.; Dhaene, T.; De Turck, F. A probabilistic ontology-based platform for self-learning context-aware healthcare applications. *Expert Systems with Applications* **2013**, *40*, 7629–7646.
8. Winters, B.; Custer, J.; Galvagno, S.M.; Colantuoni, E.; Kapoor, S.G.; Lee, H.; Goode, V.; Robinson, K.; Nakhasi, A.; Pronovost, P.; others. Diagnostic errors in the intensive care unit: a systematic review of autopsy studies. *BMJ quality & safety* **2012**, *21*, 894–902.
9. Marrone, S.; Piantadosi, G.; Fusco, R.; Petrillo, A.; Sansone, M.; Sansone, C. A Novel Model-Based Measure for Quality Evaluation of Image Registration Techniques in DCE-MRI. 2014 IEEE 27th International Symposium on Computer-Based Medical Systems. IEEE, IEEE, 2014, pp. 209–214.
10. Piantadosi, G.; Marrone, S.; Fusco, R.; Petrillo, A.; Sansone, M.; Sansone, C. Data-driven selection of motion correction techniques in breast DCE-MRI. 2015 IEEE International Symposium on Medical Measurements and Applications (MeMeA) Proceedings; IEEE: Torino, Italy, 2015; pp. 273–278.
11. Marrone, S.; Piantadosi, G.; Fusco, R.; Petrillo, A.; Sansone, M.; Sansone, C. Breast segmentation using Fuzzy C-Means and anatomical priors in DCE-MRI. Pattern Recognition (ICPR), 2016 23rd International Conference on. IEEE, 2016, pp. 1472–1477.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 November 2018 doi:10.20944/preprints201811.0339.v1

Peer-reviewed version available at *Information* **2019**, *10*, 34; doi:10.3390/info10020034

12. Della Mea, V. What is e-Health: The death of telemedicine? *Journal of Medical Internet Research* **2001**, *3*, e22.
13. Oh, H.; Rizo, C.; Enkin, M.; Jadad, A. What is eHealth?: a systematic review of published definitions. *J Med Internet Res* **2005**, *7*, e1.
14. Healy, J. Implementing e-Health in developing countries: Guidance and principles. *ICT Applications and Cyber security Division (CYB). Policies and Strategies Department.[monografía en Internet]. Bureau for Telecommunication Development International Telecommunication Union* **2008**.
15. Ball, M.J.; Lillis, J. E-health: transforming the physician/patient relationship. *International journal of medical informatics* **2001**, *61*, 1–10.
16. Eysenbach, G.; Diepgen, T.L. The role of e-health and consumer health informatics for evidence-based patient choice in the 21st century. *Clinics in dermatology* **2001**, *19*, 11–17.
17. O'donoghue, J.; Herbert, J. Data management within mHealth environments: Patient sensors, mobile devices, and databases. *Journal of Data and Information Quality (JDIQ)* **2012**, *4*, 5.
18. Berner, E.S. *Clinical Decision Support Systems*; Springer, 2007.
19. Marrone, S.; Piantadosi, G.; Fusco, R.; Petrillo, A.; Sansone, M.; Sansone, C. Automatic Lesion Detection in Breast DCE-MRI. In *Image Analysis and Processing (ICIAP)*; Springer Berlin Heidelberg, 2013; pp. 359–368.
20. Piantadosi, G.; Fusco, R.; Petrillo, A.; Sansone, M.; Sansone, C. LBP-TOP for Volume Lesion Classification in Breast DCE-MRI. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; 2015; Vol. 9279, pp. 647–657.
21. Buchanan, B. Rule based expert systems. *The MYCIN Experiments of the Stanford Heuristic Programming Project* **1984**.
22. Musen, M.A.; Middleton, B.; Greenes, R.A. Clinical decision-support systems. In *Biomedical informatics*; Springer, 2014; pp. 643–674.
23. Aronson, A.R. DiagnosisPro: the ultimate differential diagnosis assistant. *JAMA* **1997**, *277*, 426–426.
24. Khosla, V. Technology will replace 80% of what doctors do. Retrieved April 25, 2013, from http://tech.fortune.cnn.com/2012/12/04/technology-doctors-khosla/, 2012.
25. Fingberg, J.; Hansen, M.; Hansen, M.; Krasemann, H.; Iacono, L.L.; Probst, T.; Wright, J. Integrating Data Custodians in eHealth Grids–Security and Privacy Aspects. *NEC Lab Report* **2006**.
26. Papazoglou, M.P.; Traverso, P.; Dustdar, S.; Leymann, F. Service-oriented computing: State of the art and research challenges. *Computer* **2007**, *40*.
27. Spark, A. Apache Spark: Lightning-fast cluster computing, 2015.
28. Meng, X.; Bradley, J.; Yavuz, B.; Sparks, E.; Venkataraman, S.; Liu, D.; Freeman, J.; Tsai, D.; Amde, M.; Owen, S.; others. Mllib: Machine learning in apache spark. *Journal of Machine Learning Research* **2016**, *17*, 1–7.
29. Kakadia, D. *Apache Mesos Essentials*; Packt Publishing Ltd, 2015.
30. Vavilapalli, V.K.; Murthy, A.C.; Douglas, C.; Agarwal, S.; Konar, M.; Evans, R.; Graves, T.; Lowe, J.; Shah, H.; Seth, S.; others. Apache hadoop yarn: Yet another resource negotiator. Proceedings of the 4th annual Symposium on Cloud Computing. ACM, 2013, p. 5.
31. Shvachko, K.; Kuang, H.; Radia, S.; Chansler, R. The hadoop distributed file system. Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on. IEEE, 2010, pp. 1–10.
32. Dean, J.; Ghemawat, S. MapReduce: simplified data processing on large clusters. *Communications of the ACM* **2008**, *51*, 107–113.
33. Hadoop, A. Apache hadoop, 2011.
34. Meng, X.; Bradley, J.; Sparks, E.; Venkataraman, S. ML pipelines: a new high-level API for MLlib. *Databricks blog, https://databricks. com/blog/2015/01/07/ml-pipelines-a-new-high-level-api-for-mllib. html* **2015**.
35. Databricks.com. databricks. https://databricks.com/.
36. Ferrucci, D.; Levas, A.; Bagchi, S.; Gondek, D.; Mueller, E.T. Watson: beyond jeopardy! *Artificial Intelligence* **2013**, *199*, 93–105.