The Internet of Things: Opportunities and Challenges for Distributed Data Analysis

Marco Stolpe
Artificial Intelligence Group, LS 8
Department of Computer Science
TU Dortmund, 44221 Dortmund, Germany
marco.stolpe@tu-dortmund.de

ABSTRACT

Nowadays, data is created by humans as well as automatically collected by physical things, which embed electronics, software, sensors and network connectivity. Together, these entities constitute the Internet of Things (IoT). The automated analysis of its data can provide insights into previously unknown relationships between things, their environment and their users, facilitating an optimization of their behavior. Especially the real-time analysis of data, embedded into physical systems, can enable new forms of autonomous control. These in turn may lead to more sustainable applications, reducing waste and saving resources.

IoT's distributed and dynamic nature, resource constraints of sensors and embedded devices as well as the amounts of generated data are challenging even the most advanced automated data analysis methods known today. In particular, the IoT requires a new generation of distributed analysis methods.

Many existing surveys have strongly focused on the centralization of data in the cloud and big data analysis, which follows the paradigm of parallel high-performance computing. However, bandwidth and energy can be too limited for the transmission of raw data, or it is prohibited due to privacy constraints. Such communication-constrained scenarios require decentralized analysis algorithms which at least partly work directly on the generating devices.

After listing data-driven IoT applications, in contrast to existing surveys, we highlight the differences between cloud-based and decentralized analysis from an algorithmic perspective. We present the opportunities and challenges of research on communication-efficient decentralized analysis algorithms. Here, the focus is on the difficult scenario of vertically partitioned data, which covers common IoT use cases. The comprehensive bibliography aims at providing readers with a good starting point for their own work.

1. INTRODUCTION

Every day, data is generated by humans using devices as diverse as personal computers, company servers, electronic consumer appliances or mobile phones and tablets. Due to tremendous advances in hardware technology over the last few years, nowadays even larger amounts of data are automatically generated by devices and sensors, which are embedded into our physical environment. They measure, for instance,

- machine and process parameters of production processes in manufacturing,
- environmental conditions of transported goods, like cooling, in logistics,
- temperature changes and energy consumption in smart homes,
- traffic volume, air pollution and water consumption in the public sector or
- puls and bloodpressure of individuals in healthcare.

The collection and exchange of data is enabled by electronics, software, sensors and network connectivity, that are embedded into physical objects. The infrastructure which makes such objects remotely accessible and connects them, is called the *Internet of Things (IoT)*. In 2010, already 12.5 billion devices were connected to the IoT [34], a number about twice as large as the world's population at that time (6.8 billion).

The IoT revolutionizes the Internet, since not only computers are getting connected, but physical things, as well. The IoT can thus provide us with data about our physical environment, at a level of detail never known before in human history [76]. Understanding the generated data can bring about a better understanding of ourselves and the world we live in, creating opportunities to improve our way of living, learning, working, and entertaining [34]. Especially the combination of data from many different sources and their automated analysis may yield new insights into existing relationships and interactions between physical entities, their environment and users. This facilitates to optimize their behavior. Automation of the interplay between data analysis and control can lead to new types of applications that use fully autonomous optimization loops. Examples will be shown in Sect. 3, indicating their benefits.

However, IoT's inherent distributed nature, the resource constraints and dynamism of its networked participants, as well as the amounts and diverse types of data are challenging even the most advanced automated data analysis methods known today. In particular, the IoT requires a new generation of distributed algorithms which are resource-aware and intelligently reduce the amount of data transmitted and processed throughout the analysis chain.

Many surveys (for instance, [3, 43, 78, 110]) discuss IoT's underlying technologies, others [37, 81] security and privacy

issues. Data analysis' role and related challenges are only covered shortly, if at all. Some surveys [1, 12, 23, 31] mention the problem of big data analysis and propose centralized cloud-based solutions, following the paradigm of parallel high performance computing. The authors of [40], [101] and [80] take a more things-centric perspective and argue for the analysis and compression of data before its transmission to a cloud. [8] identify the need for decentralized analysis algorithms, in addition. [100] present existing applications of well-known data analysis algorithms in an IoT context, highlighting decentralized data analysis as open issue concerning infrastructure. However, they do not address an algorithmic perspective.

To the best of our knowledge, our survey is the first one dealing with differences between cloud-based and decentralized data analysis from an algorithmic perspective. In Sect. 2, we elaborate on the role of data analysis in the context of the IoT. In Sect. 3, we show, how advanced levels of data analysis could enable new types of applications. Section 4 presents the challenges for data analysis in the IoT and argue for the need of novel data analysis algorithms. Like many other authors, we see the convenience and benefits of cloud-based solutions. However, we want to move further and enable data analysis even in resource-restricted situations (Sect. 5). In Sect. 6, we argue in favor of data reduction and decentralized algorithms in highly communication-constrained scenarios which existing surveys largely neglected, so far. We focus on communication-efficient distributed analysis in the vertically partitioned data scenario, which covers common IoT use cases. Section 7 presents future research directions. Finally, we summarize and draw final conclusions. The bibliography aims at providing readers with a good starting point for their own work.

2. THE INTERNET OF THINGS

The IoT consists of physical objects (or "things") which embed electronics, software, sensors, and communication components, enabling them to collect and exchange data. Physical things are no longer separated from the virtual world, but connected to the Internet. They can be accessed remotely, i.e. monitored, controlled and even made to act.

Ideas resembling the IoT reach back to the year 1988, starting with the field of ubiquitious computing. In 1991, Mark Weiser framed his ideas for the computer of the 21st century [106]. Weiser envisioned computers being small enough to vanish from our sight, becoming part of the background, so that they are used without further thinking. Rooms would host more than 100 connected devices, which could sense their environment, exchange data and provide human beings with information similar to physical signs, notes, paper, boards, etc. Devices would need self-knowledge, e.g., of their location. Many of Weiser's original ideas can still be found in current definitions of the IoT and requirements for according devices. For example, Mattern and Floerkemeier [68] enumerate similar capabilities needed to bridge the gap between the virtual and physical world. Objects must be able to communicate and cooperate with each other, which requires addressability, unique identification, and localization. Objects may collect information about their surroundings and they may contain actuators for manipulating their environment. Objects can embed information processing, featuring a processor or microcontroller, and storage capacity. Finally, they may interface to and communicate with humans directly or indirectly. In a report by Verizon [104], the IoT is defined as a machine to machine (M2M) technology based on secure network connectivity and an associated cloud infrastructure. Things belonging to the IoT follow the so called three "A"s. They must be *aware*, i.e. sense something. They must be *autonomous*, i.e. transfer data automatically to other devices or to Internet services. They also must be *actionable*, i.e. integrate some kind of analysis or control.

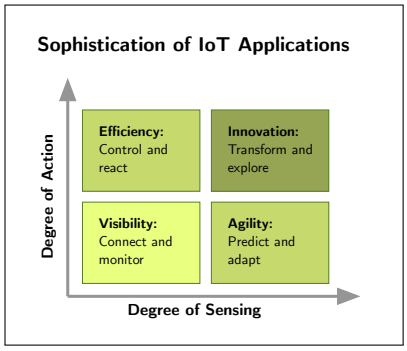


Figure 1: Sophistication levels of IoT applications [104]

The history of the IoT itself started in 1999, with the work on Radio-frequency identification (RFID) technology by the Auto-ID Center of the Massachusetts Institute of Technology (MIT) [34, 68]. The term "Internet of Things" was first literally used by the center's co-founder Kevin Ashton in 2002. In a Cisco whitepaper, Dave Evans [34] estimates that the IoT came into real existence between 2008 and 2009, when the number of devices connected to the Internet began to exceed the number of human beings on earth. Many of such devices were mobile phones, after in 2007, Steve Jobs had unveiled the first iPhone at Macworld conference. Since then, more and more devices are getting connected. It is estimated that by 2020, the IoT will consist of almost 50 billion objects [34].

The World Wide Web (WWW) fundamentally changed in at least four stages [34]. First, the web was called the Advanced Research Projects Agency Network (ARPANET) and foremost used by academia. The second stage was characterized by companies acquiring domain names and sharing information about their products and services. The "dot-com" boom may be called the third stage. Web pages moved from static to interactive transactional applications that allowed for selling and buying products online. The "social" or "experience" web marks the current fourth stage, enabling people to communicate, connect and share information. In comparison, Internet's underlying technology and protocols have gradually improved, but didn't change fundamentally. Now, connecting billions of physical things, crossing borders of entirely different types of networks poses new challenges to Internet's technologies and communication protocols. This is why the IoT was called the first evolution of the Internet [34].

As did the Internet, the IoT has the potential to change our lives in fundamental ways. Gathering and analysing data from many different sources in our environment may provide a more holistic view on the true relationships and

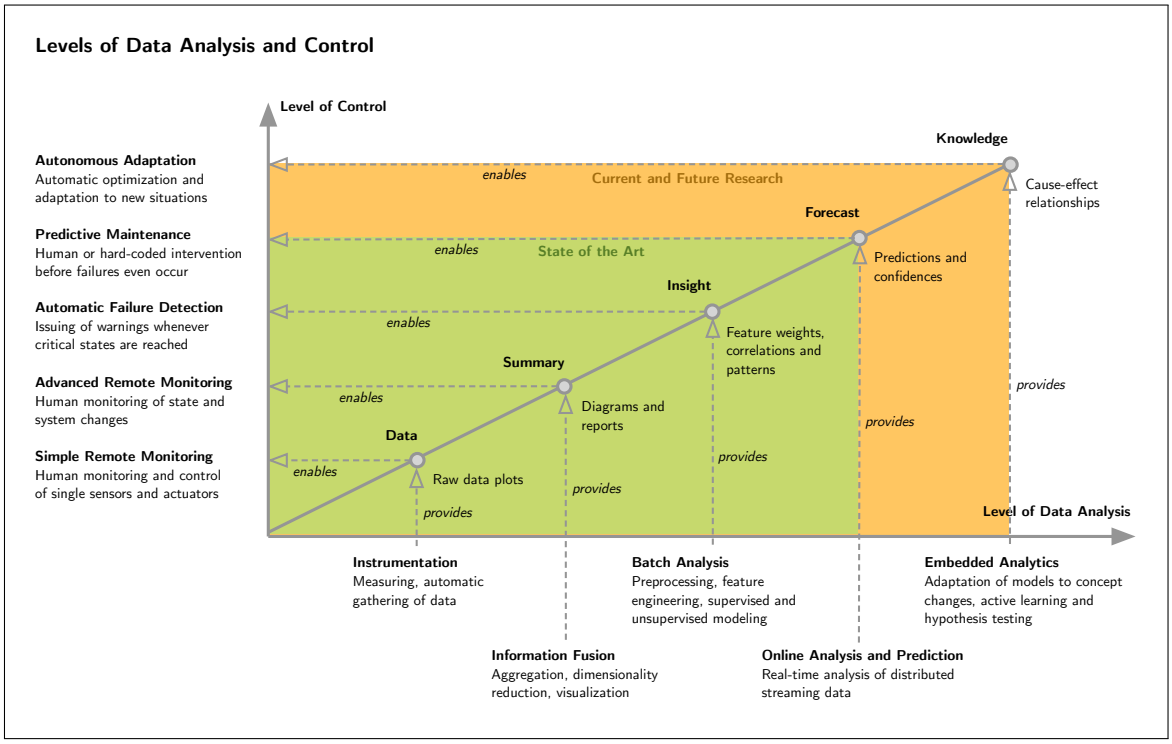


Figure 2: Relationship between data analysis and control

interactions between physical entities, enabling the transformation of raw data and information into long-term knowledge and wisdom [34]. The timely identification of current trends and patterns in the data could further support proactive behavior and planning, for instance by anticipating natural catastrophes, traffic jams, security breaches, etc. The IoT may also create new business opportunities. Potential benefits for companies are improved customer and citizen experience, better operation of machines and quality control, accelerating growth and business performance, as well as improving safety and a reduction of risk. Verizon estimates that by 2025, companies having adopted IoT technology may become 10% more profitable. Other sources predict profit increases by up to 80%. It is further estimated that the number of business to business (B2B) connections will increase from 1,2 billion in 2014 to 5,4 billion by 2020 [104]. The following section describes possible IoT applications in different sectors and points to the particular benefits that can be expected from automated data analysis.

3. DATA-DRIVEN IOT APPLICATIONS

In [25, 69], IoT applications are categorized by their level of *information and analysis* vs. their level of *automation and control*. A similar distinction is made in [104], which measures the sophistication of IoT applications by two factors, namely the *degree of action* and the *degree of sensing* (see Fig. 1). Applications falling into the lower left corner of the diagram in Fig. 1 already provide benefits given the ability to connect to and monitor physical things remotely. Giving objects a virtual identity independent of their physical location highly increases their visibility and can facilitate decision making based on smart representations of raw data. Applications located in the upper left corner of Fig. 1, in addition, use embedded actuators. Beyond pure monitoring, they enable remote control of physical things, thereby easing their management. Applications that analyse IoT generated data fall into the lower right corner of Fig. 1. Here, especially the combination of data from different physical objects and locations could provide a more holistic view and insights into phenomena that are only understood poorly, so far.

Though we agree with the previously presented categorizations, they don't show the dependency of advanced control mechanisms on data analysis. Data analysis could turn data into valuable information, which can then be utilized for building long-term knowledge and proactive decision making. Finally, merging analysis and control may lead to innovative new business models, products and services. We therefore propose the scheme in Fig. 2 which stresses the analysis. We structure the field along the dimensions of *control* and *data analysis*. The diagonal shows the milestones on the path to fully embedded analytics, which is put to good use in automatic system optimization.

The data gathered from single sensors for analysis enables simple remote monitoring applications. Here, the informed choice and placement of sensors during instrumentation depend on a well-defined analysis goal [91, 114]. Advanced applications move from the observation of single sensors to the monitoring of system and process states. This monitoring is based on the visualization of summary information obtained with the help of data analysis from multiple types of sensors and devices. The batch analysis of historical records finds correlations between features and relate them to a target value. Insights gained from this step may lead, for instance, to a better understanding of critical failure conditions and

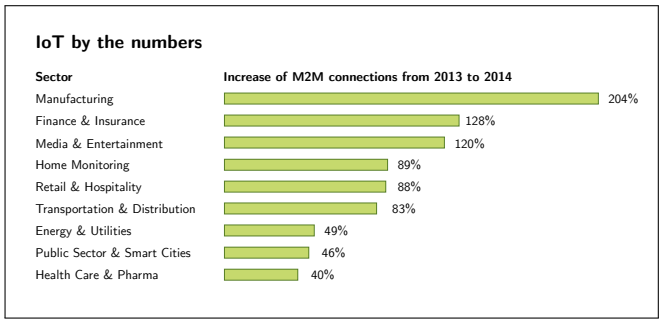


Figure 3: Increase of M2M connections in Verizon's network from 2013 to 2014 [104]

their automated detection. Prediction models derived from batch analysis may also be deployed for real-time forecasts. This is current state-of-the-art.

However, depending on the amount and rate of generated measurements, their preprocessing may become infeasable. Hence, current research focuses on distributed streaming analysis methods and the intelligent reduction of data directly at the sensors and devices themselves (see Sect. 4.3 and Sect. 5.2). Data analysis which is embedded into all parts of an IoT system will finally require the real-time derivation of models and an adaptation to changes in underlying data distributions and representations. This would in turn allow for a continuous and automated monitoring of changes in correlations. The full integration of data analysis and control introduces an automated conduction of cause-effect analysis by active testing of hypotheses, moving beyond the detection of correlations. Knowledge about causal relationships may then be used to autonomously adapt the relevant parameters in new situations. Limiting models and their use to a small selection of parameters saves memory, computing, and energy resources.

Figure 3 shows the increase of M2M connections for different business sectors in Verizon's network from 2013 to 2014. In the following, we present examples of specific IoT applications from the sectors mentioned at the beginning: Manufacturing, transportation and distribution, energy and utilities, the public sector and smart cities, as well as healthcare and pharma. We have ordered examples of each sector according to the different levels of data analysis and control as shown in Fig. 2 and have identified three main application types: Predictive maintenance, sustainable processes saving resources and quality control.

3.1 Manufacturing

The manufacturing sector supports the development of IoT by the provision of smart products. For instance, 43 million wearable bands were shipped in 2015 [20], and it is estimated that 20 million smart thermostats will ship by 2023 [74]. By 2016, smart products will be offered by 53% of manufacturers [77].

The sector not only produces devices, but also uses IoT technology itself. According to Fig. 3, the manufacturing sector is seeing the largest growth in terms of M2M connections in Verizon's network. Following the levels of Fig. 2, we now present types of industrial applications.

Simple remote monitoring applications increase visibility by embedding location-aware wireless sensors into products and wearables [104]. This allows for a continuous tracking of persons and assets, like available stock and raw materials, on- and offsite over cellular or satellite connections. [104] further mentions sensors which can detect hazards or security breaches by the instrumentation of products and wearables. Embedding sensors into production machinery will allow for the monitoring of individual machines with high granularity along the process chain. It should be added, however, that the automatic detection of such events necessarily requires an analysis and interpretation of measurements.

The aggregation of data from the same type of sensors supports the confidence in the accuracy of analysis results. Moreover, the fusion of data from different types of sensors advances remote monitoring of larger units, like systems, processes and their environment. For instance, [91] visually identify and quantify different types of productions modes in steel processing by summarizing multi-dimensional sensor data with algorithms for dimensionality reduction.

Models derived from heterogenous data sources by batch analysis may provide insights into the correlations between multiple dimensions of process parameters and a target value. According to [104], the timely identification of failure states can lead to less disruption and increase uptime in comparison to regular human maintenance visits and inspections. It should be added that once trained, data analysis models can often be made directly operational, and be used, for instance, for the automatic detection of critical patterns. For instance, learned models may be deployed early in the process for the automatic real time prediction of a product's final quality [91], allowing for timely human intervention. Here, resources might be saved by omitting further processing of already defect products. Based on human knowledge, control parameters might be adjusted such that a targeted quality level can still be reached. In the context of maintenance, the quantity to be predicted is machine wear or failure. The timely detection of anomalies and machine wear can help with reducing unplanned downtime, increasing equipment utilization and overall plant output [91, 104]. However, depending on the amount of generated data, batch analysis as well as preprocessing all data in real-time can be challenging [94]. Advanced applications therefore require the development of new kinds of data analysis algorithms (see Sect. 4.3 and Sect. 5.2).

Making data acquisition and analysis an integral part of production systems could finally allow for the long time observation of changes in correlations between process parameters and target variables. The importance of manufacturing for the adoption of IoT is emphasized by the German initiative "Industrie 4.0". It fosters the integration of production processes, IoT technology and cyber-physical systems into a so called *smart factory*. In this future type of factory, products can communicate with their environment, for instance with other products, machines and humans. In contrast to fixed structures and specifications of production processes that exist today, Reconfigurable Manufacturing Systems (RMS) derive case-specific topologies automatically based on collected data [16]. Hence, production will become more flexible and customized. Reactions to changes in customer demands and requirements may take only hours or minutes, instead of days. RMS might further support the active testing of hypotheses and targeted generation of new observations. The resulting variability of large numbers of observations

might then help with automatically distinguishing between random correlations of parameters and those the target variables truly depend on. Such knowledge could then be used for the automatic optimization and autonomous real-time adaptation of production processes and their parameters to new situations. The intelligent combination of data analysis and control can thereby lead to more sustainable systems which allow for major reductions in waste, energy costs and the need for human intervention [25, 91].

3.2 Transportation and Distribution

The sector of transportation and distribution belongs to the early adopters of IoT. Here, according to [104], important factors for the adoption of IoT technology are regulations and competition which force higher standards of efficiency and safety, as well as expectations of greater comfort and economy. From 2013 to 2014, the sector has seen a 83% increase of M2M connections in Verizon's network (see Fig. 3).

The instrumentation of vehicles enables simple remote monitoring applications that make it easier to locate and instruct fleets of cars, vans or trucks [45]. Logging driver's working hours, speed and driving behavior can improve safety and simplify compliance with regulations [104]. Customers can be regularly informed about the delivery times of anticipated goods. Even containers themselves are now equipped with boards of very restricted capacities, which open up opportunities of tracing and organizing the goods in a logistic chain of storage and delivery [103].

Another example for new types of applications is the UBER smartphone app which indicates the location of passengers calling a taxi to nearby drivers and uses surge pricing to fulfill demands for more taxis.

Advanced remote monitoring applications use data analysis to aggregate data and may provide summaries of fleet movements on a larger scale, like the average number of vehicles traveling certain routes, thereby facilitating resource planning [53].

Instrumentation allows car manufacturers deeper insights into the use of their cars. Models derived by the batch analysis of data gathered from many cars could automatically be deployed inside cars to identify or predict failure conditions. These models may also provide information about the relationships between failures and underlying causes. According to [104], such information would allow to pre-emptively issue recalls, improve designs to iron out problems, and better target new features to driver and market preferences. Intelligence built into vehicles, like proximity lane sensors, automatic breaking, head lamps, wipers, and automated emergency calls can increase road safety [104].

Advanced applications, like autonomously driving vehicles [5], require the embedded real-time analysis of data directly inside the vehicle. In addition, information sent by nearby infrastructure, like traffic signals, traffic signs, street lamps, road works or local weather stations might be taken into account (see also Sect. 3.4). For navigation, vehicles may remotely access current information on street maps.

At a larger scale, data gathered from many vehicles and infrastructure could be analysed and used to instruct vehicles beyond their individual driving decisions. [54] developed a sophisticated distributed analysis of local data from vans of a fleet, which allows to manage the overall fleet. Work orders can be allocated in real-time more efficiently, adopting to drivers, reacting to order changes, or other events. The effects on cutting fuel costs, leading to more sustainable vehicles and distribution systems has been shown [72]. Similarly, through timely diagnostics, predictive analytics, and the elimination of waste in fleet scheduling, the rail industry is looking to achieve savings of 27 billion dollars globally over 15 years [35].

3.3 Energy and Utilities

In the sector of energy distribution, IoT applications range from telematics for job scheduling and routing, to bigger ones extending the life of electricity infrastructure [104]. According to Fig. 3, the energy sector has seen an estimated growth of 49% in the number of M2M connections from 2013 to 2014.

Concerning remote monitoring, the energy sector was the first to introduce SCADA (supervisory control and data acquisition). Smart meters increase visibility by providing more granular data. Thereby they reduce the incovenience and expense of manual meter readings or estimated bills. Further, advanced remote monitoring provides more accurate views of capacity, demand and supply over different smart homes, made possible by visualizing summary information obtained from data analysis [55, 65, 116]. Based on such information, sustainability may be improved through better resource planning and cutting energy theft. According to [104], in 2014, 94 million smart meters were shipped worldwide and it is predicted that by 2022, the number of smart meters will reach 1.1 billion. One target of the European Union is to replace 80% of meters by smart meters by 2020, in 28 member countries.

Beyond monitoring applications, the batch analysis of data from smart homes may help with giving recommendations for saving energy and enable more sophisticated energy management applications [116]. Oil and gas companies can cut costs and increase efficiency by early predicting the failure of artificial components, local weather conditions, and the automated start up and shutdown of equipment [104]. On a larger scale, the smart grid connects assets in the generation, transmission and distribution infrastructures. Especially in recent years, energy use has become harder to predict, due to a decentralization of energy production. The prediction of wind power [99] and photovoltaic power [108] is important in order to better understand grid utilization. Data analysis may increase efficiency and optimize the infrastructure [55]. The embedded real-time analysis of data could enable even more sustainable distributed energy generation models in which highly autonomous systems react dynamically to changes in energy demand and distribute energy accordingly.

3.4 Public Sector

In the public sector, M2M connections have grown by 46% from 2013 to 2014 according to Fig. 3. It is estimated that by 2050, 66% of humans will live in urban areas [102] and 75% of world's energy use is taking place in cities [104]. The IoT promises the delivery of more effective services to citizens, like citizen's participation, controlling crime, the protection of infrastructure, keeping power and traffic running, and building sustainable developments with limited resources [104]. The IoT thus enables municipal leaders to make their communities safer and more pleasant to live, and to deal better with demographic changes [104].

The instrumentation of cities with sensors may lead to more

sustainable resource usage by simple remote monitoring applications. For instance, currently it takes 20 minutes on average to find a parking space in London [104] and 30% of congestion in cities is caused by people looking for a parking space [84]. The smart city of Santander [86] has instrumented, among others, parking lots. Their space utilization could be tracked and provided as information to smart phone apps. Advanced applications may also identify trends and anomalies in parking data [115]. Similar tracking apps could support car-sharing or unattended rental programs that offer on-demand access to vehicles by the hour [104]. More advanced remote monitoring applications could indicate the crowdedness of neighboring cities by aggregating data with the help of data analysis. Using real-time analysis, they might as well give direct recommendations, for instance which city to visit for more relaxed shopping.

Resource savings can also be expected from a more sustainable management of water. IBM offers an intelligent software for water management that uses data analysis for visualization and correlation detection [50]. According to IBM, the software helps to manage pressure, detect leaks, reduce water consumption, mitigate sewer overflow and allows for a better management of water infrastructure, assets and operations.

Currently, up to 40% of municipal energy costs come from street lighting [109]. The European Union has set a target to reduce CO_2 emissions of professional lighting by 20 million tons by 2020 [104]. Predictive models obtained through data analysis enable smart streetlights that automatically adjust their brightness according to the expected volume of cars and weather conditions. In a case study it was shown that the city of Lansing, Michigan, could thereby cut the energy and maintenance costs of street lighting by 70% [88, 104].

Further resources might be saved by using more intelligent transportation and traffic systems. Predicting traffic flow on the basis of past data that has been measured by sensors in the streets offers drivers an enhanced routing. The German government estimated a daily fuel consumption in Germany due to traffic jams of 33 millions of liter, a waste of time in the range of 13 million hours and concludes that traffic jams are responsible for an economic loss of 259 million Euro per day. For instance, the SCATS system [83] provides traffic flow data for different junctions throughout Dublin city. Simple remote monitoring can provide data about the current traffic flow to individual drivers by plotting counts of cars on a digital street map. The batch analysis of traffic data could help with determining factors causing traffic jams, which in turn might be used by traffic managers to adapt the street network accordingly. For the City of Dublin, traffic forecast derived from a spatio-temporal probabilistic graphical model, was exploited for smart routing [62]. In the future, such recommendations may be as well given to autonomously driving vehicles (see also Sect. 3.2).

Embedding data analysis everywhere in a city and combining the data from multiple heterogenous systems and other cities may even provide larger value. Such combination could provide a holistic view of everything, like energy use, traffic flows, crime rate and air pollution [104]. Correlations and relationships between seemingly unrelated variables are not necessarily obvious. For instance, according to the broken windows theory, the prevention of small crimes such as vandalism helps with preventing more serious crimes. However, critics state that other factors have more influence on crime rate. Up to now, such theories are hard to test and validate, since studies conducted by humans can only focus on a limited number of influence factors and might be biased. The instrumentation of many different cities and areas could increase the number of observations and help with obtaining more objective and statistically significant results. Long time observation of many different variables and active hypothesis testing, for instance by giving recommendations to city planners, may help with the detection of causes that underly phenomena. The insights gained may then enable better policy decisions.

3.5 Healthcare and Pharma

According to Fig. 3, healthcare has seen the smallest growth in M2M connection from 2013 to 2014. Similarly, Gartner estimates that it will take between five and 10 years for a full adoption of the IoT by health care. This slow adoption rate may be explained by strict requirements for keeping data of patients private and secure [42], with the IoT posing many challenges for privacy and security (see also Sect. 4). Despite such difficulties, the number and possible impact of IoT applications in healthcare is large.

The instrumentation of healthy citizens as well as patients, devices or even whole hospitals with different kinds of sensors enables different kinds of remote monitoring applications. It starts with consumer-based devices for personal use. In two years, there will be 80 million wearable health devices [42], like fitness trackers and smart watches. New kinds of devices are able to monitor not only the number of steps taken or calories, but also pulse rate, blood pressure, or blood sugar levels. The aggregation of these kinds of different information requires data analysis [36]. Monitoring might promote healthy behavior through increased information and engagement [57]. In addition, physicians may get more holistic pictures of their patients' life styles, which eases diagnosis [18].

Monitoring can be done remotely and continuously in real time, beyond office visits, with patients staying at home [18, 25, 70]. Emergencies can be detected early, like with breath pillows for children or Ion mobility spectrometry combined with multi-capillary columns (MCC/IMS) that can give immediate information about the human health status or infection threats [47]. In the case of chronic illnesses, practitioners get early warning of conditions that would lead to unplanned hospitalizations and expensive emergency care [25, 42,45,57]. Monitoring alone could reduce treatment costs by a billion dollars annually in the US [25]. According to [42], estimates show a 64% drop in hospital readmissions for heart failure patients whose blood pressure and oxygen saturation levels were monitored remotely. Similarly, at-risk elderly individuals may longer stay in their own homes. Here, remote monitoring can reassure loved ones by detecting falls or whether an indivdual got out of bed in the morning, or whether an individual took his or her medicine [57].

Monitoring may as well help with drug management and the detection of fraudulent drugs in the supply chain, by incorporating RFID tags in medication containers and finally embedding technology in the medication itself [45]. In hospitals, medical equipment like MRIs and CTs can be connected and remotely monitored, helping with maintenance, replenishing supplies and reducing expensive downtime [42].

While conditions based on a few measurements may be detected automatically based on hard-wired rules, the detec-

tion of more complex patterns necessarily requires the analysis of data.
Data analysis is also needed, if we want to identify critical patterns in patient's vital parameters [51, 60] or in movements through hospitals and optimize flow [42]. The analysis of multi-dimensional data is necessary for discovering dependencies between many variables, like, e.g., the duration of treatments and waiting times at other wards. Data analysis provides doctors with insights of scientific value, taking the data gathered by many individuals as population-based evidence [57]. Clinical and nonclinical data of larger population samples may help to understand the unique causes of a disease. Finally, data analysis that was directly embedded into devices like electrocardigrams (ECG) or wireless electrocardiograms (WES) could help with the detection of emergency cases in real-time [24].

4. DATA ANALYSIS CHALLENGES

The previous section has given many examples of applications in diverse sectors, showing that advanced levels of control not only require the instrumentation of devices, but also an analysis of the acquired data. These examples support our view expressed in Fig. 2 that it is data analysis which enables advanced types of control. Unfortunately, the IoT poses new challenges to data analysis. The following sections present problems in terms of security and privacy, technical issues as well as algorithmic challenges which require research on new types of data analysis methods.

4.1 Security and Privacy

Despite IoT's anticipated positive effects, it also poses risks for our security and privacy. Especially sectors that deal with highly personalized information, such as healthcare (see Sect. 3.5), require according means for the secure and privacy-preserving processing of data. Apart from having to make existing data analysis code more secure, analysis can as well provide solutions to decrease existing threats.

Security. The biggest security risk of IoT stems from its biggest benefit, namely the connection of physical things to a global network. In the past, security breaches were mostly restricted to the theft and manipulation of data *about* physical entities. However, the IoT allows for a direct control of the physical entities *themselves*, many of which belonging to critical infrastructures in sectors previously mentioned. Without security measures, malware like viruses could easily spread through many of IoT's connected networks, potentially resulting in disasters at a global scale [32, 37].
Data analysis algorithms can be made secure by design. However, existing code bases weren't necessarily designed and implemented with security in mind. In the past, algorithms could be expected to run mostly in environments which weren't publicly accessed. Further, the way how data has been input into analysis software was relatively controlled. With the IoT, analysis code will run on devices directly exposed to an open network environment and is thus suspectible to malicious hacking attempts. It will be much harder to ensure that data originates from trustworthy sources and is in appropriate format. Hackers might gain access to sensors and other embedded devices [32, 37, 81], or install rogue devices that interfere with existing network traffic [81]. Hence, it becomes more and more important to make data analysis code more robust by penetration testing [33] and differentiate hacking attempts from usual sensor failure. Also, legal liability frameworks must be established for algorithms whose decisions are fully automated [25].
At the same time, data analysis might provide solutions for the automatic detection or even prevention of security breaches. For instance, outlier and novelty detection algorithms which examine deviations from normal behavior have already been used successfully in fields like intrusion or malware detection [10, 17].

Privacy. Another of IoT's challenges is the protection of citizens' privacy. As Mark Weiser already stated in 1991, "hundreds of computers in every room, all capable of sensing people near them and linked by high-speed networks, have the potential to make totalitarianism up to now seem like sheerest anarchy" [106]. Since it became known that intelligence agencies of democratic states are spying at other friendly states and their citizens [96], the topic of privacy has developed an especially high brisance. It also plays a large role in business sectors where data is highly personalized. For instance, data in healthcare must be especially protected.
One problem is that with small embedded devices vanishing from our sight, people might not even recognize that data about them is getting acquired. Further, it may not be entirely clear how data given away will be combined later on and what can then be derived from it. For instance, as research on learning from label proportions [79, 93] suggests, information that seems harmless all by itself, like public election results, may become problematic once it is combined with data from other sources, such as social web sites.
It is important to mention, however, that several of the aforementioned benefits from data analysis can be achieved without highly personalized data [41]. For instance, disease research based on population-based evidence (see Sect. 3.5) would yield the same results with anonymized observations. If that doesn't suffice and enough samples are present, data can further be aggregated to guarantee k-anonymity [95]. Related is the problem of learning from label proportions [79, 93]. Where more privacy is needed, the challenge consists of developing distributed analysis algorithms that derive a model without exchanging individualized records between different networked nodes (for instance, see [27]).

4.2 Technical Challenges

Technical challenges of IoT mainly concern networking technology, devices interoperability, as well as increasing the lifetime and range of wireless battery-powered devices. Here, we list the technical problems that every application of data analysis has to face.

Data Understanding. One envisioned scenario for the analysis of IoT generated data is that as people connect new devices to the IoT, their data is automatically getting analysed, together with the data of other already existing devices. Data analysis being successful, however, depends much on the correct preprocessing of data, which in turn depends on the types and ranges of features of observations. This information can be estimated from the data. However, it can be difficult to assess the quality of such estimations without ground truth. For instance, outlier detection al-

gorithms may indicate measurements which occur only seldom. However, without additional background knowledge provided by experts, it is impossible to determine automatically if values are still inside physically meaningful ranges or caused by sensor failure. Similarly, peak detection algorithms might wrongly identify noise as relevant patterns. These problems could easily be solved if manufacturers made their sensors and embedded devices queryable and provided meta data, e.g. meaningful ranges and noise levels of theirs sensors.

Standardization. The ability to query sensors and devices for meta information requires standardized protocols. A similar standardization is needed for the exchange of raw data. Especially in industry, closed systems with proprietary data formats complicate the exchange of data between distributed components and make automated data analysis unnecessarily difficult [91]. Similarly important would be a standardization of user interfaces for data analysis tools. As Mark Weiser already noted in [106], technology becomes unobtrusive once its user interfaces are as uniform and consistent as possible. In contrast, today the user interface of operating systems and applications often is their most distinguishing property and therefore a unique selling point. Hence, a wide adoption of common standards requires that profits made from IoT technology outweigh potential losses caused by the lacking individualization of products.

Porting existing code bases. As Sect. 4.1 already discussed, existing code bases for data analysis must be made more robust to operate in hostile network environments. In addition, as more and more data analysis algorithms can be expected to run directly on embedded and mobile devices, existing code and related libraries need to be ported to these platforms. The implementation language of choice for embedded devices is C/C++. In contrast, much data analysis code is written in Java and Python, whose virtual machines and interpreters require too many resources to run on small embedded devices like sensors. Currently, the same algorithms must therefore be implemented in many different versions, making the reuse of existing code more difficult. Beyond modification of existing code bases, the IoT poses several challenges that require research on new algorithms, as described in the next section.

4.3 Algorithmic Challenges

Manual inspection of IoT generated data is possible only in simple cases. Normally, since the amount of data generated by single sensors becomes too high, the analysis needs to be fully automated. Further, the combination of data from many heterogenous sources leads to high-dimensional datasets that cannot be easily visualized or examined by humans.

Automated data analysis methods have been developed in the fields of signal processing and computer vision [29], statistics [46], artificial intelligence [82], machine learning [71], data mining [44] and databases [39], to name just some text books. Among them are sophisticated methods that can generalize over raw data, deriving *models* that describe patterns and relationships which statistically hold on expectation also for unseen observations. Such methods will be called *learning algorithms* in the following. Unsupervised learning algorithms find general patterns and relationships in the data. Supervised algorithms find such patterns in relation to a specified target value, which at best should be given as label for each observation. The difficulty in both cases is that the model must be derived only from a given finite *sample* of the data, while the probability distribution generating the data is unknown (for a more formal definition of the problem, see [46]). Many learning algorithms assume the sample to be given as a single batch which can be processed in a random access fashion, potentially making several passes over the data. Observations are assumed to have a relatively homogenous structure and fixed representation.

The IoT poses new challenges to data analysis. At the data generating side, devices are often highly resource-constrained in terms of CPU power, available main memory, external storage capacity, energy and available bandwidth. Algorithms working at the data generating side must take these constraints into account. Also the underlying data distribution may change which is known as *concept drift* [117]. For instance, due to wear, the accuracy of sensors may decrease. At the receiving side, e.g. a data center, the combination of data from many different sources may create huge masses of heterogenous data. It is estimated that in total, the IoT will generate 4.4 trillion GB by 2020 [75]. Hence, the problem consists of having to analyse *big data* [67, 76], which is characterized by large *volume* (terabytes or even petabytes of data), *heterogenity* (different sources and formats) and *velocity* (speed of generated data). High volume and velocity prohibit several passes over the data, and thus require new types of algorithms. In addition to the big data problem, the analysis of IoT data are distributed and asynchronous. Just to illustrate an effect of this particular setting, let us look at IoT devices dynamically entering or leaving the network. This contradicts an assumption underlying almost all data analysis approaches, namely that the representation of observations, e.g. the number of features, does not change over time.

5. DISTRIBUTED DATA ANALYSIS

The requirements of algorithms for the analysis of IoT generated data are largely determined by the hardware and network environment in which they are expected to run. Depending on volume and rate of data generation, as well as the particular analysis problem, data must either be already preprocessed and analyzed at the generating side, on network middleware or sent to a data center. Each scenario comes with its own set of advantages and disadvantages, constraints and particular challenges. Based on specifications found on websites of cloud providers and manufacturers, we have compiled a list of computing environments and device's properties for a quick and easy comparison in Fig. 4.

The current focus is on the centralization of data in the cloud and its analysis by high performance computing [19, 23, 31, 43, 76]. Cloud computing allows for highly scalable distributed systems that solve tasks in parallel by means of virtualization. Virtual instances of nodes in a network are independent from the particular physical nodes they run on. Hence, new instances can easily be added and removed depending on current computational demands. Computation follows the paradigm of parallel computing in so far as modern frameworks shield programmers as much as possible from the intricate details of distributed systems. For

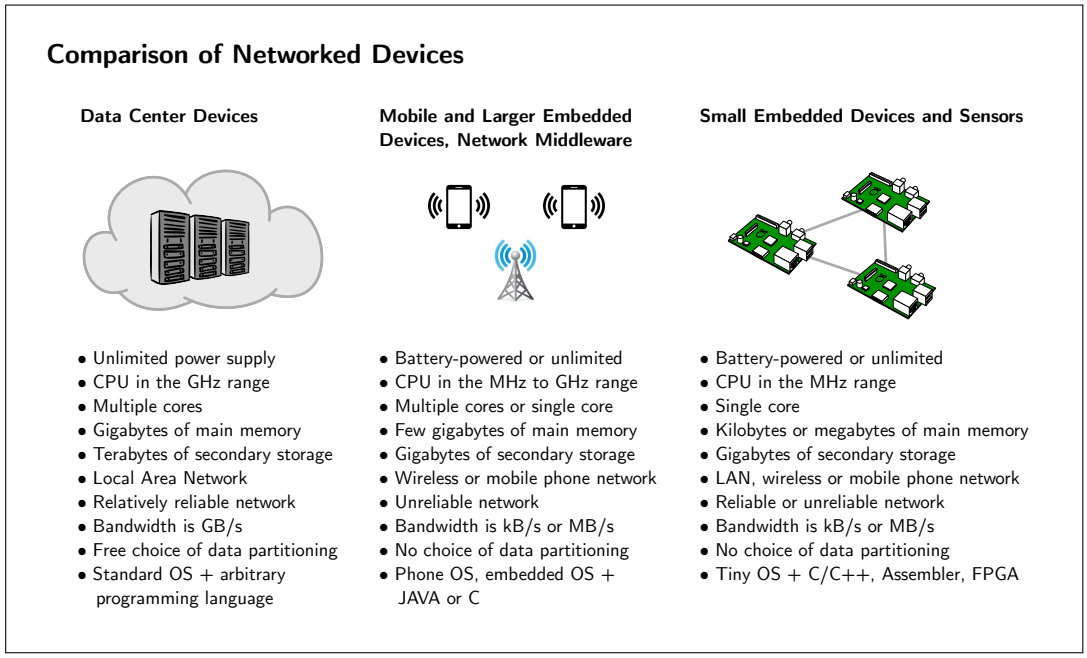


Figure 4: Comparison of computing environments and device types

instance, the scheduling and execution of code, the creation of threads or processes, synchronization as well as message passing are handled automatically. Failures that can occur in distributed systems are taken care of by redundancy and the automatic rescheduling of processes. The main task for programmers is to divide their problem into smaller subproblems which can be worked on in parallel. How and where code is executed is mostly transparent, giving the impression of a single big machine instead of many nodes.

As more and more devices are getting connected, existing network hardware and infrastructure will no longer suffice to handle the expected network traffic [19, 25, 26, 70, 76]. Whenever the rate of data generation is higher than available bandwidth, data must be analysed on the generating devices themselves or at least be reduced *before* transmission into the cloud [40, 80, 101]. In the following, algorithms that process or analyse data directly where it is acquired will be called *decentralized*. In case they need another node for coordination, data and computation are at least splitted between local nodes and the coordinator. Decentralized algorithms which need no coordinator and exchange information only with local peer nodes will be called *fully decentralized*. Ideally, decentralized analysis algorithms should exchange less information than all data between nodes.

The next section presents the ideas and constraints of current cloud-based data analysis approaches in more detail, while the following section discusses the need for decentralized data analysis algorithms in more communication-constrained scenarios.

5.1 Data Centers and Cloud Computing

One option for the analysis of IoT generated data is its centralization at a data center. Cloud computing solutions are offered by different service providers. They allow for an easy and cost-efficient upscaling of computing and storage resources. Depending on the rate of data generation, there exist two different models of data processing: Data may either be stored and analysed as a batch, or it must be processed directly as a stream.

Batch analysis. Huge data masses which do not fit in one server require the distribution of data over different connected storage devices. This is, for instance, accomplished by saving chunks of arriving data in a distributed file system such as HDFS [85]. Once the data is stored, it can be analysed as a batch by distributed algorithms that solve tasks cooperatively. Each machine in a data center may have multiple cores, which algorithms can exploit for parallel execution. CPUs are in the gigahertz (GHz) range and main memory has several gigabytes. Machines are usually connected in a local area network (LAN) where connections are relatively reliable. Technologies such as Infiniband and 100 Gigabit Ethernet allow for high bandwidths which are comparable to direct main memory accesses. Reading from dynamic random access memory (DRAM) can be about one order of magnitude faster than reading from external storage mediums, like solid-state drives (SSDs). A reorganization of data would therefore be an expensive operation. Hence, it is desirable to read data from disk only once. This can be achieved by moving code to the machine storing the data and executing it locally.

The distributed batch analysis of data is currently supported by different frameworks. Hadoop [107] is a popular framework. It follows the map and reduce paradigm known from functional programming, where the same code is executed on different parts of the data and the results are then merged. Map and reduce is especially well-suited for problems that are data parallel. This means that tasks can work independently from each other on different chunks of data, reading it only once, without synchronization or managing state. The

paradigm lends itself well for data analysis algorithms which process subsets of observations or features only once. Some algorithms for counting, preprocessing and data transformation fall into this category.
More advanced data analysis algorithms, especially learning algorithms, often require the combination of data from different subsets. They also need to make several passes over the data, and synchronize shared model parameters. For instance, the k-Means clustering algorithm [64] repeatedly assigns observations to a globally maintained set of centroids. Similarly, many distributed optimization algorithms used in data analysis maintain a globally shared set of model parameters (see also [13]). In map and reduce, distributed components are assumed to be stateless. One way to maintain state between iterations would be to access, for instance, a database server which is external to the Hadoop framework. However, this would require the unnecessary and repeated transmission of state over the network. For the implementation of stateful components, lower level frameworks like the Message Passing Interface (MPI) [2] or ZeroMQ [49] are usually better suited. These frameworks allow for long running stateful components and full control over which data is to be sent over the network.
Distributed variants of well-known data analysis algorithms, like k-Means clustering [64] and random forests [15], have been implemented in the Apache mahout [98] framework that works on top of Hadoop. However, the framework contains only few algorithms, as research on distributed data analysis algorithms for high performance computing is still ongoing.

Analysis of streaming data. Whenever batch processing isn't fast enough to provide an up-to-date view of the data, it must be processed as a stream [9,26]. The Lambda architecture by Marz [67] is a hybrid of batch and stream processing. The batch layer regularly creates views on historical data. The speed layer processes current data items which come in while batch jobs are running, and creates up-to-date views for this data. Both views are combined at a service layer, which provides a single view on the data to users. A disadvantage of the Lambda architecture is that algorithms must be designed and implemented for different layers. Kreps [58] therefore proposed the Kappa architecture, in which all data is treated as a stream.
Several frameworks support the development of streaming algorithms (for one framework and an overview, see [9]). Related analysis algorithms are still an active area of research [38] and are currently implemented in different frameworks [7, 30, 97].

The centralization of all data in the cloud offers several benefits. The often complicated network infrastructure needed for distributed computing as well as the corresponding machines are fully managed by the provider. Due to providers' expert knowledge, security risks might decrease. Customers pay only for those services they really use, such that it becomes easier and less costly to accomodate for spikes in network traffic. As long as the data analysis algorithms to be executed and their components can be fully parallelized, scalability is just a matter of adding new machines.
However, the centralization of all data also poses risks for privacy and may have disadvantages. In the case of data theft, all data may suddenly become accessible. Further, the cloud itself poses a single point of failure. Whenever data is generated at a higher rate than can be transmitted, either due to a limited bandwidth or high latency, the cloud can become a bottleneck for real-time analysis and control. Such cases require the local processing and reduction of data directly at the data generating side, as argued for in the next section.

Table 1: Data transfer rates of different technologies

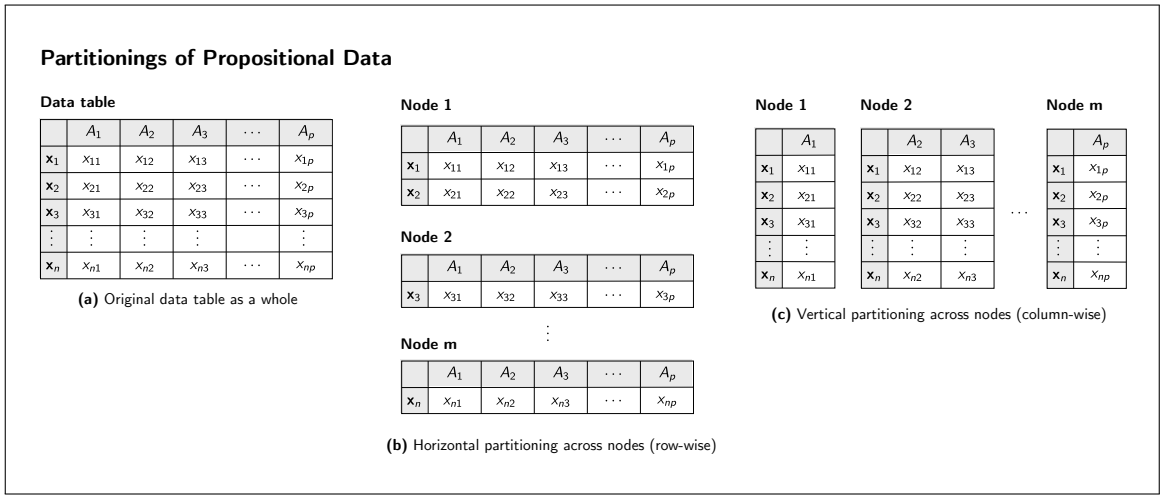
Technology	**Rate**		**Type**
EDGE	237.0	kB/s	Mobile Phone
UMTS 3G	48.0	kB/s	Mobile Phone
LTE	40.75	MB/s	Mobile Phone
802.15.4 (2.4 GHz)	31.25	kB/s	Wireless
Bluetooth 4.0	3.0	MB/s	Wireless
IEEE 802.11n	75.0	MB/s	Wireless
IEEE 802.11ad	900.0	MB/s	Wireless
Solid-state drive (SSD)	600.0	MB/s	Storage
eSATA	750.0	MB/s	Peripheral
USB 3.0	625.0	MB/s	Peripheral
VDSL2	12.5	MB/s	Broadband
Ethernet	1.25	MB/s	Local Area
Gigabit Ethernet	125.0	MB/s	Local Area
100 Gigabit Ethernet	12.5	GB/s	Local Area
Infiniband EDR 12x	37.5	GB/s	Local Area
PC4-25600 DDR4 SDRAM	25.6	GB/s	Memory

5.2 Communication-constrained Scenarios

A central analysis of IoT generated data requires its transmission over a network. However, due to technical limitations, the transmission of *all* data to a central location, like a data center, is not always possible. Either the data generating devices themselves are highly communication-constrained, or the available bandwidth is too limited. Moreover, there exist cases where privacy concerns, security concerns, business competition or political regulations prohibit the centralization of all data.

Communication-constrained devices. One of mobile devices' biggest constraint is that they are battery powered. Devices having much less computational power, like embedded devices or smart sensors, can be battery powered as well, even if they aren't mobile. Sending and receiving data is known to be one of the most energy draining operations on mobile devices [22] and smart sensors [63]. Hence, communication must be traded off against computation.

Limitations of bandwidth. There exist several scenarios in which the available bandwidth does not suffice to transmit all data to a central location. IoT generated data may stem from devices that are connected wirelessly. Table 1 shows typical transfer rates for different kinds of network technologies and bus systems. It becomes apparent that wireless networks provide much lower bandwidths than LANs which are used in data centers. For instance, ZigBee networks based on IEEE 802.15.4, a specification for personal area networks consisting of small, low-power digital radios, have a data transmission rate of only 31.25 kB/s. Mobile devices, like smartphones or tablets, are relatively powerful in



(a) Original data table as a whole

(b) Horizontal partitioning across nodes (row-wise)

(c) Vertical partitioning across nodes (column-wise)

Figure 5: Common types of data partitioning

terms of computation and available main memory (see also Fig. 4). They easily may generate data at higher rates than can be transmitted over mobile telephone interfaces. Other applications, like those in earth science [112] or telescopes in physics [11], produce masses of data whose transmission over satellite connections is in the range of years. Masses of data are also generated by high throughput applications, like Formula One racing [89], which require a real-time analysis of large amounts of data [26]. Similarly, analysis and control in manufacturing can have real-time constraints [91,94]. In cases where reaction times lie in the range of a few seconds, it seems risky to send production parameters first into the cloud for preprocessing and analysis, which then computes an answer. Depending on latency, which can be high with Internet based services, the answer may come too late. Finally, bandwidth becomes more limited with more network participants. With the IoT, those will likely increase as more and more devices are getting connected to the same network segments [76]. According to [70], "how to control the huge amount of data injected into the network from the environment is a problem so far mostly neglected in the IoT research".

Privacy concerns and regulations. Privacy concerns and regulations may entirely prohibit the transmission of data to a central location. Or, privacy-preserving algorithms may transmit data, but not the original records. Further, network usage might be constrained by political or business regulations, such that data cannot be centralized. Other issues concern security and fail-safe operation. Centralized systems pose single points of failure. The more control is depending on data and its analysis, the more important it is to guarantee its delivery. In the cloud computing scenario, service provider and client may secure their end points, but usually have no control over the transmission of packets in between. A smart factory sending all its data into the cloud, depending on a timely analysis for real-time operation, might come to a complete standstill in case of a network failure. Even if the cloud is not available, continuous local operation should at least be possible.

In all of the aforementioned cases, data must be directly analysed on the generating devices themselves and be reduced before transmission (see also [8, 26, 40, 80]). For instance, as shown in [63], the reduction of data before transmission with the help of autogressive models reduced the energy consumption of smart sensors (MEMS) by factors up to 11. Similar reductions could be achieved with edge mining [40], whose authors argue purely in favor of local data preprocessing. However, local transformations and models may not suffice to capture dependencies between highly correlated measurements from different sensors. In such cases, decentralized algorithms are needed which build a global model based on messages exchanged between peer nodes or with a coordinator node. Such algorithms will necessarily need to be designed differently from distributed algorithms running in a data center. There, network technology allows for transfer rates resembling those of main memory accesses. Moreover, it may be freely decided how data is getting stored and partitioned across machines. New storage and compute nodes may be dynamically added to the network, based on demand. However, on the data generating side, the kind of data partitioning as well as the network structure are usually application dependent and given as fixed. Especially the type of data partitioning can have a large influence on learning and the amount of data that needs to be communicated, as shown in the following section.

6. TYPES OF DATA PARTITIONING

Data for learning is often given as a sample S of n observations, i.e. $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. For the following discussion, w.l.o.g. it is assumed that observations are represented in *propositional* form, i.e. described by a finite set of p different *features* $A_1, \ldots, A_p$ (also called *attributes*). Feature values are stored in columns of a data table, with one observation per row (see Fig. 5a). In distributed settings, data from this table may be spread across nodes in two different ways [21].

Horizontal partitioning. In the *horizontally partitioned data* scenario (see Fig. 5b), data about observation, i.e. rows of the data table, are distributed across nodes $j = 1, \ldots, m$.

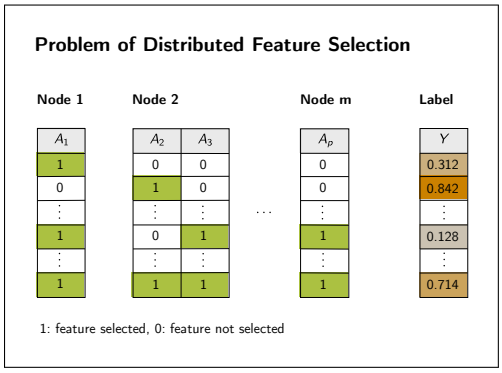


Figure 6: Which features provide most information about the target concept?

All observations share the *same features*.

Horizontally partitioned sets of observations may be seen as skewed subsamples of a dataset that would result from centralizing and merging all observations. Hence, the distributed learning task consists of building a global model from such local samples, with as few communication between nodes as possible. Observations may be assumed to be independent and identically distributed, which for instance is exploited by learning algorithms that merge summary information independently derived from each subsample. In general, there exist many distributed learning algorithms for the scenario (for instance [14, 27, 52, 65]), though only few algorithms are truly suited for small devices (for a more detailed treatment, see [6,90]). Communication costs for the scenario are well understood in the sense that bounds have been established for different classes of learning problems [4, 113]. For instance, [4] show that a distributed perceptron, which is a linear classifier, can find a consistent hypothesis in at most $O(k(1+\alpha/\gamma^2))$ rounds of communication, k being the number of nodes, supposed that data is α-well-spread and all points have margin at least γ with the separating hyperplane.

An example task for learning in the horizontally partitioned data scenario is link quality prediction in wireless sensor networks (WSNs). We may assume that factors influencing link quality are the same across different wireless sensor nodes, i.e. recorded features provide information about the same underlying concept to be learned. However, the distributions of observations may differ for different parts of the network. For instance, in certain parts the link quality could be better than in other parts. The question is how to learn a global model which represents the distribution over all observations across nodes, without having to transfer all observations to a central node.

Vertical partitioning. In the *vertically partitioned data* scenario (see Fig. 5c), feature values of observations, i.e. columns of the data table, are distributed across nodes $j = 1, \ldots, m$. Shared is only the index column, such that it is known which features belong to which observation. This might require a continuous tracking of objects, which in the IoT would be realized through globally unique identifiers for each entity.

The columns distributed over nodes constitute subspaces of the whole instance space. These subspaces and their individual components (e.g. features), in supervised learning including the target label, have a dependency structure that is usually unknown before learning. Learning in the scenario may thus be seen as a combinatorial problem of exponential size: Which subset of features provides the most information about the target concept (see also Fig. 6)? In supervised learning, this is also known as the *feature selection* [87] problem, whereas in unsupervised learning similar problems occur in *subspace clustering* [59]. Several techniques have been developed to tackle the exponential search space [56]. Most of them are highly iterative and assume that features can be freely combined with each other. In a decentralized setting, however, such combination requires the costly transmission of column information between nodes in each iteration step and is thus prohibited. Hence, current approaches [28,61,92] circumvent such problems by making explicit assumptions on the conditional joint dependencies of features, given the label.

In the context of the IoT, learning in the vertically partitioned data scenario is relevant and common. The problem occurs whenever a state or event is to be detected or predicted, based on feature values assessed at different nodes. What exactly constitutes a single observation then is application dependent. A common use case are spatio-temporal prediction models, which use measurements of devices at different locations. Measurements may be related to each other by the time interval in which they occur. The following list gives examples of applications:

- In manufacturing, one is interested in predicting the final product quality as early as possible [91,94], based on process parameters and measurements at different production steps. Similarly, the optimization of process flow could benefit from a prediction of the time it takes to assemble a product, based on the current filling of queues and machine parameters at different locations on a shop floor. In both cases, a single observation consists of features, like sensor measurements and machine parameters, that are distributed and assessed at different locations. Depending on the granularity of control to be achieved, predictions must be either given after minutes, seconds or maybe also milliseconds. The more time-constrained the application, the more it might benefit from decentralized local processing.
- Products are assembled from parts delivered by different suppliers [105]. Optimal planning and scheduling of assembly steps depend on a correct and continuous estimation of parts' delivery times. Those again are determined by production and transportation parameters of individual suppliers. For instance, the delivery of a particular part might be delayed due to the maintenance of a single production unit at one supplier. Assembly time of a product is thus a global function depending on local information (features) from different suppliers, i.e. observations for learning this function are vertically partitioned. Even if it was technically feasable to centralize the raw production and transportation data from all suppliers for analysis, it would be unnecessary if the global function depended only on a few local features. Moreover, due to privacy concerns, it is unrealistic that suppliers would provide raw data about their processes. Hence, a decentralized

algorithm is needed that derives a global model from local data, at the same time preserving privacy.

- The smart grid requires a continuous prediction of energy demand [55,65], based on local information about energy usage at different smart homes [116]. Here, observations might represent the whole state of the energy grid, and consist of vertically partitioned features at different locations describing local states. Instead of centralizing raw meter readings from ten thousands of households, communication could be spared by an aggregation of local data or a combination of predictions from locally trained models.

- Centralized traffic management systems analyse traffic based on data from a hard-wired mesh of distributed presence sensors [83]. While easy to design, centralized systems pose a single point of failure in case of an emergency. With the addition of new sensors, they may become a bottleneck, due to limited bandwidth. Further, the maintenance of hard-wired sensors can be expensive in case of failure, due to required construction work. A more decentralized system could consist of cheap wireless sensors. Those may be attached to existing infrastructure, like traffic lights, signs and street lights. Traffic lights may then adjust themselves, based on the prediction of traffic flow at neighboring junctions. The flow measurements at each individual junction can be interpreted as vertically partitioned features of a single observation describing the current state of all sensors. The learning task is to derive prediction models from these distributed flow measurements, without transmission of all data to a central server [92].

- In healthcare, diagnoses of illnesses depend on many factors, like a patient's health care records, parents' illnesses and current health parameters such as puls, blood pressure, measurements from a blood sample, an electroencephalogram or other specialized information. With IoT technology, even more data becomes available through fitness trackers or dieting apps (see also Sect. 3.5). The features describing a single patient are thus distributed over different locations, like several physicians, medical centers, and now even devices or social websites. The centralization of all data poses a threat to patients' privacy. Hence, the learning task is to derive a global model for diagnosis from local data, without transmission of raw data between locations. The features of diagnoses from different geographical locations over certain time intervals could then be combined to predict, for instance, epidemics and their spread at a larger scale (see also [73]). Again, the features from different locations over the same time intervals constitute vertically partitioned observations.

7. RESEARCH QUESTIONS

The number of communication-efficient distributed data analysis methods for the vertically partitioned is much smaller than those for horizontally partitioned data. There are many open research questions, which mainly concern the relationship between accuracy and communication costs. Therefore, we first define how communication costs are measured and what it means for an algorithm to be communication-efficient. Then, an overview of typical components that vertically distributed algorithms may consist of is given. It is shown that the schema is general enough to cover common designs of distributed algorithms. Finally, open issues and research questions are formulated that concern communication-efficient learning.

7.1 Communication Costs and Efficiency

In most publications on distributed data analysis, *communication costs* are the total payload transmitted measured in bits, i.e. excluding meta data, like packet headers. The authors of [40] argue for a measurement of communication costs by the number of transmitted packets. Although the number of packets in certain cases might be a more exact measure than the payload in bits, it is highly dependent on chosen network protocols and the underlying network technology. Similar to measuring the run-time of algorithms in seconds, it would make the comparison of results from different publications very difficult. A fair comparison would require building the exact same network with the same hardware and configuration. A solution could be network simulators, however, there doesn't seem to exist a commonly agreed on standard between different scientifc communities. At least for batch transmissions of data, the number of packets to be sent is proportional to the payload in bits. From there, we follow the argumentation in [40] that a reduction of packets may reduce congestion and collisions on networks with large amounts of traffic. This in turn reduces the number of acknowledgements and retransmissions, which should enable better use of available bandwidth (i.e. higher transmission rates or more network participants).

Central analysis requires the transmission of all data (or at least all preprocessed data) to the coordinator node. We define a learning method to be *communication-efficient* if less data than the whole dataset (optionally after local preprocessing) is exchanged between local nodes and an optional coordinator node. Method A is called *more communication-efficient* than method B, if A is communication-efficient and its communication costs are less than those of B.

The amount of data communicated per observation during learning may differ from the amount communicated when making an actual prediction. It should be noted that in the vertically partitioned data scenario, at least *some* data must be communicated for detecting a global state or predicting a global event. Further, the supervised learning of local models may require the transmission of label information from a coordinator. This is different from a horizontal partitioning of data, where each local node contains all the necessary information (i.e. feature values and often also the label).

7.2 Distributed Setting and Components

Figure 7 gives an overview of the setting in the vertically partitioned data scenario and the distributed components that algorithms may be designed of. Given are $m+1$ networked nodes $j = 0, \dots m$, where nodes $1, \dots, m$ are called *local nodes* and $j = 0$ denotes a *coordinator node*. No assumptions are made on network topology or technology. Further, "local" and "coordinator" are to be understood as *roles* that physical nodes can have, and may change depending on context.

Each local node acquires raw values, like sensor measurements. Those may be locally preprocessed and transformed

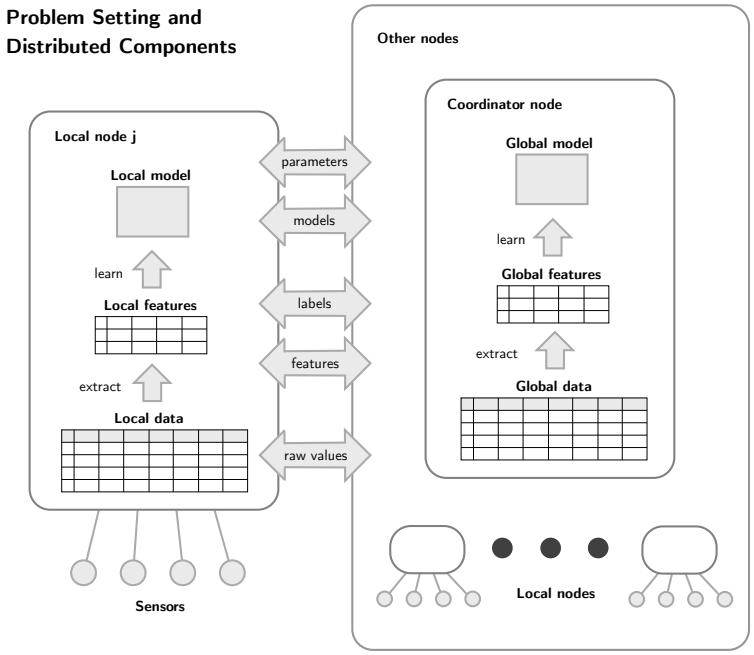


Figure 7: Problem setting and distributed components

into features for learning. It is assumed that the features of the same observations are vertically partitioned across the local nodes. Distributed components of learning algorithms may do further local calculations on such features, and might build or update local models. Once or iteratively, depending on algorithm, local nodes will either transmit raw values, features, models or predictions of such models to other nodes, which in turn may preprocess the received data, and do further calculations on them, like build or update a global model, fuse predictions, etc. The setting as described is general enough to cover the following common approaches for designing distributed algorithms:

Central analysis Each local node transmits all of its raw values to a coordinator node for further analysis. This may include the stages of preprocessing, feature extraction and model building. This is in principle what cloud-based data processing proposes [19,23,31,43,76]: While the coordinator may consist itself of distributed components and solve the analyis problem in parallel, from the perspective of local nodes it looks like a single machine where all data is getting centralized. The design is not decentralized, as data and processing aren't split between local nodes and coordinator node, but all processing is done at the coordinator node.

Local preprocessing, central analysis Local nodes preprocess raw values and transform them into a representation for learning. The representations are sent to a coordinator, which builds a global model based on them. While according to our former definition, this design is decentralized, its form is very rudimentary, as most of the processing is still done at the coordinator. Depending on the processing capabilities of local nodes and the particular learning task, such a design might be the only viable option. The design fits ideas mentioned in [40, 80, 101], whose authors' propose to reduce data locally before sending it to the cloud for analysis. Privacy-preserving Support Vector Machine (SVM) algorithms like [66, 111] also follow this design, but are not necessarily communication-efficient.

Model consensus Local nodes iteratively try to reach consensus on a set of parameters among each other (peer-to-peer), or on a set of parameters they share with a coordinator node. At the end, each local node (or only the coordinator) has a global model. As [13] demonstrate, many existing analysis problems can be cast into a consensus problem and then be solved, for instance, with the Alternating Direction Method of Multipliers (ADMM). Algorithms of this sort are working fully decentralized, but are working iteratively and may transmit more than the original data, depending on their convergence properties.

Fusion of local models Each local node preprocesses its own data and builds a local model on it. Such models are then transmitted to a coordinator node or to peer nodes, which fuse them to a global model. These algorithms are working decentralized, as data and the load of processing are shared among all nodes. In the vertically partitioned data scenario, using a global model usually requires the transmission of feature values of observations whenever a prediction is to be made. An example algorithm would be [48].

Fusion of local predictions Each local node preprocesses its own data and builds a local model on it. Whenever a prediction is to be made, only the predictions

are transmitted from local nodes, and fused at the coordinator or other peer nodes according to a fusion rule. This could be, for instance, a majority vote over predictions. Local nodes each transmit only one value during prediction, but a fusion rule may not be as accurate as a global model, depending on data distribution and learning task. Examples would be [61, 92].

While aforementioned approaches are common, there exist hybrids also covered by the setting shown in Fig. 7. For instance, in [28] local models are used to detect local outliers, which are then checked against a global model that was derived from a small sample of all data.

According to our previous definition, the examples of distributed algorithms given above all learn from vertically partitioned data in a decentralized fashion. However, not all are communication-efficient. Apart from the two mentioned privacy-preserving SVMs which might send more data than the whole dataset, the model consensus based algorithms may send more data as well, depending on the number of iterations during optimization. The design of communication-efficient decentralized algorithms in the vertically partitioned data scenario leaves many open research questions of which some are presented in the following.

7.3 Open Questions

Despite first successes in the development of communication-efficient algorithms for the vertically partitioned data scenario, there are still many open research questions left:

- Data analysis knows many different kinds of tasks, like dimensionality reduction, classification, regression, clustering, outlier detection or frequent itemset mining. How does the task influence communication costs when the data is vertically partitioned? And how does the design change with the task?
- As first results suggest, the accuracy of communication-efficient algorithms in the vertically partitioned data scenario very much depends on the data being analysed. What influence have different data distributions on the communication costs and accuracy of algorithms? How is the design of algorithms affected?
- What are bounds on communication, i.e. how much information must be at least and at most communicated to learn successfully from vertically partitioned data?
- How can the supervised learning of local models be made more communication-efficient in cases where labels do not reside on the local nodes, but must first be transmitted to them? For instance, how can we learn from aggregated label information?
- Many existing data analysis algorithms can easily work on different numbers of observations, but expect the number of features to be fixed. How can algorithms that work on observations with features from different sensors deal with the dynamic addition and removal of sensors, i.e. features?

Beyond those questions, there are open issues concerning distributed data analysis algorithms in general, i.e. also those that work on horizontally partitioned data or in the cloud. For instance, methods for feature selection, the optimization of hyper parameters and validation are highly iterative and work on different subsets of features and observations in each iteration. How can we adapt these algorithms in such a way that the same data isn't repeatedly sent over the network or read from external storage? As the previous questions demonstrate, there is still a lot of research to do before data analysis and the IoT will become seamlessly integrated.

8. SUMMARY

After a short introduction to the IoT, it was argued for data analysis being an essential part of it. By giving examples from different sectors, it was shown that already remote monitoring applications may benefit from a summarization of data with the help of data analysis. Complex applications require more advanced and autonomous control mechanisms. These in turn depend on advanced data analysis methods, like those that can analyse data in real-time, adapt to changing concepts and representations and test hypotheses actively. Beyond security, privacy and technical problems, especially algorithmic challenges need to be tackled before such advanced applications will become a reality.

Distributed cloud-based algorithms follow the paradigm of parallel high performance computing. The cloud might seem like the most convenient and powerful solution for the analysis of IoT generated big data, which is expected to have large volume, high velocity and high heterogenity. However, without substantial advances in network technology, bandwidth will become more and more scarce with each new device getting connected. The transmission of all data into the cloud can already be infeasable, due to limited energy, bandwidth, high latency or due to privacy concerns and regulations. Communication-constrained applications require decentralized analysis algorithms which at least partly work directly on the devices generating the data, like sensors and embedded devices. A particularly challenging scenario is that of vertically partitioned data, which covers common IoT use cases, but for which not many data analysis algorithms exist so far. The main research question is how to design communication-efficient decentralized algorithms for the scenario, while at the same time preserving the accuracy of their centralized counterparts.

Several works achieved impressive resource savings by reducing data with the help of analysis directly on embedded devices and sensors. In the field of data analysis, research on communication-efficient decentralized algorithms is active, as several given citations demonstrate. It seems surprising that many other surveys focus mostly on cloud-based analysis solutions, ignoring the up-coming challenges of communication-constrained IoT applications. We hope to have closed this gap by our work and providing a comprehensive bibliography. We think that in the future IoT, cloud-based and decentralized data analyis solutions will coexist and complement each other.

9. ACKNOWLEDGEMENTS

This work has been supported by the DFG, Collaborative Research Center SFB 876 (http://sfb876.tu-dortmund.de/), project B3.

10. REFERENCES

[1] C. Aggarwal, N. Ashish, and A. Sheth. The Internet of Things: A Survey From The Data-Centric Perspective. In C. C. Aggarwal, editor, *Managing and Mining Sensor Data*. Springer, Berlin, Heidelberg, 2013.

[2] Argonne National Laboratory. The Message Passing Interface (MPI) standard. `http://www.mcs.anl.gov/research/projects/mpi/`, 2015. [Online; accessed 2015-12-15].

[3] L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A survey. *Comput. Netw.*, 54(15):2787–2805, 2010.

[4] M.-F. Balcan, A. Blum, S. Fine, and Y. Mansour. Distributed Learning, Communication Complexity and Privacy. In *JMLR: Workshop and Conference Proceedings, 25th Annual Conference on Learning Theory*, 2012.

[5] W. Bernhart and M. Winterhoff. Autonomous Driving: Disruptive Innovation that Promises to Change the Automotive Industry as We Know It. In J. Langheim, editor, *Energy Consumption and Autonomous Driving: Proc. of the 3rd CESA Automotive Electronics Congress*. Springer, 2016.

[6] K. Bhaduri and M. Stolpe. Distributed Data Mining in Sensor Networks. In C. Aggarwal, editor, *Managing and Mining Sensor Data*. Springer, Berlin, Heidelberg, 2013.

[7] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. MOA: Massive Online Analysis. *J. Mach. Learn. Res.*, 11:1601–1604, Aug. 2010.

[8] S. Bin, L. Yuan, and W. Xiaoyi. Research on Data Mining Models for the Internet of Things. In *Proc. of the Int. Conf. on Image Analysis and Signal Processing (IASP)*, pages 127–132, 2010.

[9] C. Bockermann. *Mining Big Data Streams for Multiple Concepts*. PhD thesis, TU Dortmund, Dortmund, Germany, 2015.

[10] C. Bockermann, M. Apel, and M. Meier. Learning SQL for Database Intrusion Detection Using Context-Sensitive Modelling. In U. Flegel, , and D. Bruschi, editors, *Proc. of the 6th Int. Conf. on Detection of Intrusions and Malware (DIMVA)*, pages 196–205. Springer, Berlin, Heidelberg, 2009.

[11] C. Bockermann, K. Brügge, J. Buss, A. Egorov, K. Morik, W. Rhode, and T. Ruhe. Online Analysis of High-Volume Data Streams in Astroparticle Physics. In *Proc. of the European Conf. on Machine Learning (ECML), Industrial Track*. Springer, 2015.

[12] A. Botta, W. de Donato, V. Persico, and A. Pescapé. Integration of Cloud computing and Internet of Things: A survey. *Future Gener. Comp. Sy.*, 56:684–700, 2016.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan. 2011.

[14] J. Branch, C. Giannella, B. Szymanski, R. Wolff, and H. Kargupta. In-network outlier detection in wireless sensor networks. *Knowl. Inf. Sys.*, 34(1):23–54, 2012.

[15] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[16] M. Brettel, N. Friederichsen, M. Keller, and M. Rosenberg. How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective. *Int. Journ. of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 8(1):37–44, 2014.

[17] A. Buczak and E. Guven. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*, 2015.

[18] N. Bui and M. Zorzi. Health Care Applications: A Solution Based on the Internet of Things. In *Proc. of the 4th Int. Symp. on Applied Sciences in Biomedical and Communication Technologies*, ISABEL '11, pages 131:1–131:5. ACM, 2011.

[19] D. Burrus. The Internet of Things is Far Bigger Than Anyone Realizes. `http://www.wired.com/insights/2014/11/the-internet-of-things-bigger/`, 2014. [Online; accessed 2016-02-16].

[20] Canalys. Wearable band shipments set to exceed 43.2 million units in 2015. `http://www.canalys.com/newsroom/wearable-band-shipments-set-exceed-432-million-units-2015`, 2014. [Online; accessed 2016-04-04].

[21] D. Caragea, A. Silvescu, and V. Honavar. Agents that learn from distributed dynamic data sources. In *Proc. of the Workshop on Learning Agents*, 2000.

[22] A. Carroll and G. Heiser. An Analysis of Power Consumption in a Smartphone. In *Proc. of the 2010 USENIX Conf. on USENIX Ann. Technical Conf. (USENIXATC)*, USA, 2010. USENIX Association.

[23] F. Chen, P. Deng, J. Wan, D. Zhang, A. Vasilakos, and X. Rong. Data Mining for the Internet of Things: Literature Review and Challenges. *Int. J. Distrib. Sen. Netw.*, 2015:12:12–12:12, Jan. 2015.

[24] M. Chen, Y. Ma, J. Wang, D. Mau, and E. Song. Enabling Comfortable Sports Therapy for Patient: A Novel Lightweight Durable and Portable ECG Monitoring System. In *IEEE 15th Int. Conf. on e-Health Networking, Applications and Services (Healthcom)*, pages 271–273, 2013.

[25] M. Chui, M. Löffler, and R. Roberts. The Internet of Things. `http://www.mckinsey.com/insights/high_tech_telecoms_internet/the_internet_of_things`, Mar. 2010. [Online; accessed 2016-02-16].

[26] F. Combaneyre. Understanding Data Streams in IoT. http://www.sas.com/en_us/whitepapers/understanding-data-streams-in-iot-107491.html, 2015. [Online; accessed 2016-02-23].

[27] K. Das, K. Bhaduri, and H. Kargupta. A local asynchronous distributed privacy preserving feature selection algorithm for large peer-to-peer networks. *Knowledge and Information Systems*, 24(3):341–367, 2009.

[28] K. Das, K. Bhaduri, and P. Votava. Distributed Anomaly Detection Using 1-class SVM for Vertically Partitioned Data. *Stat. Anal. Data Min.*, 4(4):393–406, Aug. 2011.

[29] E. Davies. *Computer and Machine Vision: Theory, Algorithms, Practicalities*. Academic Pr, Inc., 2012.

[30] G. De Francisci Morales and A. Bifet. SAMOA: Scalable Advanced Massive Online Analysis. *J. Mach. Learn. Res.*, 16(1):149–153, Jan. 2015.

[31] M. Díaz, C. Martín, and B. Rubio. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *Journal of Network and Computer Applications*, 2016.

[32] J. Dixon. Who Will Step Up To Secure The Internet of Things? http://techcrunch.com/2015/10/02/who-will-step-up-to-secure-the-internet-of-things/, 2015. [Online; accessed 2016-02-16].

[33] P. Engebretson. *The Basics of Hacking and Penetration Testing*. Elsevier/Syngress, 2nd edition, 2013.

[34] D. Evans. The Internet of Things – How the Next Evolution of the Internet Is Changing Everything. https://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, Apr. 2011. [Online; accessed 2015-11-19].

[35] P. Evans and M. Annunziata. Industrial Internet: Pushing the Boundaries of Minds and Machines. http://www.ge.com/docs/chapters/Industrial_Internet.pdf, 2012. [Online; accessed 2016-04-04].

[36] T. Fawcett. Mining the Quantified Self: Personal Knowledge Discovery as a Challenge for Data Science. *Big Data*, 3(4):249–266, Jan. 2016.

[37] D. Fletcher. Internet of Things. In M. Blowers, editor, *Evolution of Cyber Technologies and Operations to 2035*, pages 19–32. Springer International Publishing, 2015.

[38] J. Gama. *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, 1st edition, 2010.

[39] H. Garcia-Molina, J. Ullman, and J. Widom. *Database Systems: The Complete Book*. Pearson Education Limited, 2nd edition, 2013.

[40] E. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic. Edge Mining the Internet of Things. *IEEE Sensors Journal*, 13(10):3816–3825, 2013.

[41] F. Gianotti and D. Pedreschi, editors. *Mobility, Data Mining and Privacy*. Springer, 2007.

[42] J. Glaser. How The Internet of Things Will Affect Health Care. http://www.hhnmag.com/articles/3438-how-the-internet-of-things-will-affect-health-care, Jun. 2015. [Online; accessed 2016-02-23].

[43] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Gener. Comput. Syst.*, 29(7):1645–1660, Sep. 2013.

[44] J. Han and M. Kamber. *Data Mining*. Morgan Kaufmann, 2nd edition, 2006.

[45] B. Harpham. How the Internet of Things is changing healthcare and transportation. http://www.cio.com/article/2981481/healthcare/how-the-internet-of-things-is-changing-healthcare-and-transportation.html, Sep. 2015. [Online; accessed 2016-02-16].

[46] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

[47] A.-C. Hauschild, T. Schneider, J. Pauling, K. Rupp, M. Jang, J. Baumbach, and J. Baumbach. Computational Methods for Metabolomic Data Analysis of Ion Mobility Spectrometry Data - Rviewing the State of the Art. *Metabolites*, 2(4):733–755, 2012.

[48] C. Heinze, B. McWilliams, and N. Meinshausen. DUAL-LOCO: Preserving privacy between features in distributed estimation. In *Proc. of the 19th Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, JMLR: Workshop and Conference Proceedings, 2016.

[49] P. Hintjens. *ZeroMQ*. O'Reilly, USA, 2013.

[50] IBM. IBM Intelligent Water: Water management software with analytics for improved infrastructure and operations. http://www-03.ibm.com/software/products/en/intelligentwater, 2016. [Online; accessed 2016-04-01].

[51] M. Imhoff, R. Fried, U. Gather, and V. Lanius. Dimension Reduction for Physiological Variables Using Graphical Modeling. In *AMIA 2003, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 8-12, 2003*, 2003.

[52] M. Kamp, M. Boley, D. Keren, A. Schuster, and I. Scharfman. Communication-Efficient Distributed Online Prediction by Decentralized Variance Monitoring. In T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, editors, *Proc. of the European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery (ECML/PKDD)*, pages 623–639. Springer, 2014.

[53] H. Kargupta, R. Bhargava, K. Liu, M. Powers, P. Blair, S. Bushra, J. Dull, K. Sarkar, M. Klein, M. Vasa, and D. Handy. VEDAS: A Mobile and Distributed Data Stream Mining System for Real-Time Vehicle Monitoring. In *Proc. of the SIAM Int. Conf. on Data Mining (SDM)*, chapter 28, pages 300–311. 2004.

[54] H. Kargupta, K. Sarkar, and M. Gilligan. MineFleet: an overview of a widely adopted distributed vehicle performance data mining system. In *Proc. of the 16th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 37–46, 2010.

[55] A. Khan, A. Mahmood, A. Safdar, Z. Khan, and N. Khan. Load forecasting, dynamic pricing and DSM in smart grid: A review. *Renew. Sust. Energ. Rev.*, 54:1311–1322, 2016.

[56] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[57] R. Krawiec, J. Nadler, P. Kinchley, E. Tye, and J. Jarboe. No appointment necessary: How the IoT and patient-generated data can unlock health care value. `http://dupress.com/articles/internet-of-things-iot-in-health-care-industry/`, Aug. 2015. [Online; accessed 2016-02-16].

[58] J. Kreps. Questioning the Lambda Architecture. `http://radar.oreilly.com/2014/07/questioning-the-lambda-architecture.html`, 2014. [Online; accessed 2015-12-15].

[59] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering High-dimensional Data: A Survey on Subspace Clustering, Pattern-based Clustering, and Correlation Clustering. *ACM Trans. Knowl. Discov. Data*, 3(1):1:1–1:58, Mar. 2009.

[60] V. Lanius and U. Gather. Robust online signal extraction from multivariate time series. *Comput. Stat. Data An.*, 54(4):966–975, 2010.

[61] S. Lee, M. Stolpe, and K. Morik. Separable Approximate Optimization of Support Vector Machines for Distributed Sensing. In P. Flach, T. D. Bie, and N. Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7524 of *LNCS*, pages 387–402, Berlin, Heidelberg, 2012. Springer.

[62] T. Liebig, N. Piatkowski, C. Bockermann, and K. Morik. Predictive Trip Planning - Smart Routing in Smart Cities. In *Proc. of the Workshop on Mining Urban Data at the Int. Conf. on Extending Database Technology*, pages 331–338, 2014.

[63] J. Long, J. Swidrak, M. Feng, and O. Buyukozturk. Smart Sensors: A Study of Power Consumption and Reliability. In E. Wee Sit, editor, *Sensors and Instrumentation, Volume 5: Proc. of the 33rd IMAC, A Conf. and Exposition on Structural Dynamics*, pages 53–60. Springer, 2015.

[64] J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability*, pages 281–297, 1967.

[65] R. Mallik and H. Kargupta. A Sustainable Approach for Demand Prediction in Smart Grids using a Distributed Local Asynchronous Algorithm. In *Proc. of the Conf. on Data Understanding (CIDU)*, pages 1–15, 2011.

[66] O. Mangasarian, E. Wild, and G. Fung. Privacy-preserving Classification of Vertically Partitioned Data via Random Kernels. *ACM Trans. Knowl. Discov. Data*, 2(3):12:1–12:16, Oct. 2008.

[67] N. Marz and J. Warren. *Big Data - Principles and best practices of scalable realtime data systems*. Manning, 2014.

[68] F. Mattern and C. Floerkemeier. From the Internet of Computers to the Internet of Things. In K. Sachs, I. Petrov, and P. Guerrero, editors, *From Active Data Management to Event-based Systems and More*, pages 242–259. Springer-Verlag, Berlin, Heidelberg, 2010.

[69] M. May, B. Berendt, A. Cornuejols, J. Gama, F. Giannotti, A. Hotho, D. Malerba, E. Menesalvas, K. Morik, R. Pedersen, L. Saitta, Y. Saygin, A. Schuster, and K. Vanhoof. Research Challenges in Ubiquitous Knowledge Discovery. In Kargupta, Han, Yu, Motwani, and Kumar, editors, *Next Generation of Data Mining (NGDM)*, pages 131–151. CRC Press, 2009.

[70] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516, 2012.

[71] T. Mitchell. *Machine Learning*. Mcgraw-Hill Education Ltd, 1997.

[72] K. Morik, K. Bhaduri, and H. Kargupta. Introduction to data mining for sustainability. *Data Min. Knowl. Disc.*, 24(2):311–324, Mar. 2012.

[73] R. Moss, A. Zarebski, P. Dawson, and J. McCaw. Forecasting influenza outbreak dynamics in melbourne from internet search query surveillance data. *Influenza and Other Respiratory Viruses*, page n/a, Feb. 2016.

[74] Navigant Research. Shipments of Smart Thermostats Are Expected to Reach Nearly 20 Million by 2023. `https://www.navigantresearch.com/newsroom/shipments-of-smart-thermostats-are-expected-to-reach-nearly-20-million-by-2023`, 2014. [Online; accessed 2016-04-04].

[75] Oracle Corporation. Energize Your Business with IoT-Enabled Applications. `http://www.oracle.com/us/dm/oracle-iot-cloud-service-2625351.pdf`, 2015. [Online; accessed 2016-02-16].

[76] Oracle Corporation. Unlocking the Promise of a Connected World: Using the Cloud to Enable the Internet of Things. `http://www.oracle.com/us/solutions/internetofthings/iot-and-cloud-wp-2686546.pdf`, 2015. [Online; accessed 2015-12-15].

[77] Oxford Economics. Manufacturing Transformation: Achieving competetive advantage in changing global marketplace. `http://www.oxfordeconomics.com/Media/Default/Thought%20Leadership/executive-interviews-and-case-studies/PTC/Manufacturing%20Transformation%20130607.pdf`, 2013. [Online; accessed 2016-04-04].

[78] D. Partynski and S. Koo. Integration of Smart Sensor Networks into Internet of Things: Challenges and Applications. In *Proc. of the IEEE Int. Conf. on Green Computing and Communications (GreenCom) and IEEE Internet of Things (iThings) and IEEE Cyber, Physical and Social Computing (CPSCom)*, pages 1162–1167, 2013.

[79] G. Patrini, R. Nock, T. Caetano, and P. Rivera. (Almost) No Label No Cry. In *Advances in Neural Information Processing Systems (NIPS)*, number 27, pages 190–198. Curran Associates, Inc., 2014.

[80] Y. Qin, Q. Sheng, N. Falkner, S. Dustdar, H. Wang, and A. Vasilakos. When things matter: A survey on data-centric internet of things. *J. Netw. Comput. Appl.*, 64:137–153, 2016.

[81] R. Roman, J. Zhou, and J. Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266–2279, 2013.

[82] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2013.

[83] SCATS. Sydney Coordinated Adaptive Traffic System. `http://www.scats.com.au/`, 2013. [Online; accessed 2015-08-19].

[84] D. Shoup. Free Parking or Free Markets. *Cato Unbound - A Journal of Debate*, 2011. [Online; accessed 2016-04-04].

[85] K. Shvachko, H. K., S. Radia, and R. Chansler. The Hadoop Distributed File System. In *IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 1–10, 2010.

[86] SmartSantanderSantander. Future Internet Research & Experimentation. `http://www.smartsantander.eu`, 2016. [Online; accessed 2016-04-01].

[87] U. Stanczyk and L. Jain, editors. *Feature Selection for Data and Pattern Recognition*. Studies in Computational Intelligence. Springer, 2015.

[88] W. Stephenson. IntelliStreets: Digital Scaffolding for 'Smart' Cities. `http://www.huffingtonpost.com/w-david-stephenson/intellistreets_b_1242972.html`, 2012. [Online; accessed 2016-04-04].

[89] J. Stierwalt. Formula 1 and HANA: How F1 Racing is Pioneering Big Data Analytics. `http://jeremystierwalt.com/2014/01/29/formula-1-and-hana-how-f1-racing-is-pioneering-big-data-analytics/`, 2014. [Online; accessed 2016-02-16].

[90] M. Stolpe, K. Bhaduri, and K. Das. Distributed Support Vector Machines: An Overview. In *Solving Large Scale Learning Tasks: Challenges and Algorithms*, volume 9580 of *LNCS*. 2016. [to appear].

[91] M. Stolpe, H. Blom, and K. Morik. Sustainable Industrial Processes by Embedded Real-Time Quality Prediction. In J. Lässig, K. Kerstin, and K. Morik, editors, *Computational Sustainability*, volume 9570 of *LNCS*, pages 207–251. Springer, Berlin, Heidelberg, 2016.

[92] M. Stolpe, T. Liebig, and K. Morik. Communication-efficient learning of traffic flow in a network of wireless presence sensors. In *Proc. of the Workshop on Parallel and Distributed Computing for Knowledge Discovery in Data Bases (PDCKDD)*, CEUR Workshop Proceedings, page (to appear). CEUR-WS, 2015.

[93] M. Stolpe and K. Morik. Learning from Label Proportions by Optimizing Cluster Model Selection. In *Proc. of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases*, volume 3, pages 349–364, Berlin, Heidelberg, 2011. Springer-Verlag.

[94] M. Stolpe, K. Morik, B. Konrad, D. Lieber, and J. Deuse. Challenges for Data Mining on Sensor Data of Interlinked Processes. In *Proceedings of the Next Generation Data Mining Summit (NGDM) 2011*, 2011.

[95] L. Sweeney. K-anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, Oct. 2002.

[96] J. Tapper. Obama administration spied on German media as well as its government. 2015. [Online; accessed: 2016-03-30].

[97] The Apache Software Foundation. Apache Flink: Scalable Batch and Stream Data Processing. `http://flink.apache.org/`, 2015. [Online; accessed 2015-12-15].

[98] The Apache Software Foundation. mahout. `http://mahout.apache.org/`, 2015. [Online; accessed 2016-03-30].

[99] N. Treiber, J. Heinermann, and O. Kramer. Wind Power Prediction with Machine Learning. In J. Lässig, K. Kersting, and K. Morik, editors, *Computational Sustainability*, volume 9570 of *LNCS*. Springer, 2016.

[100] C.-W. Tsai, C. Lai, M. Chiang, and L. Yang. Data Mining for Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials*, 16(1):77–97, 2014.

[101] C.-W. Tsai, C.-F. Lai, and A. Vasilakos. Future Internet of Things: open issues and challenges. *Wirel. Netw.*, 20(8):2201–2217, 2014.

[102] United Nations. World Urbanization Prospects. `http://esa.un.org/unpd/wup/Publications/Files/WUP2014-Report.pdf`, 2014. [Online; accessed 2016-04-04].

[103] A. K. R. Venkatapathy, A. Riesner, M. Roidl, J. Emmerich, and M. ten Hompel. PhyNode: An intelligent, cyber-physical system with energy neutral operation for PhyNetLab. In *Proc. of the Europ. Conf. on Smart Objects, Systems and Technologies, Smart SysTech*, 2015.

[104] Verizon. State of the Market: The Internet of Things 2015. http://www.verizonenterprise.com/state-of-the-market-internet-of-things/, 2015. [Online; accessed 2015-10-22].

[105] C. Wang, Z. Bi, and L. D. Xu. IoT and Cloud Computing in Automation of Assembly Modeling Systems. *IEEE T. Ind. Inform.*, 10(2):1426–1434, 2014.

[106] M. Weiser. The Computer for the 21st Century. *Sci. Am.*, 265(9), 1991.

[107] T. White. *Hadoop: The Definitive Guide*. O'Reilly, USA, 2nd edition, 2011.

[108] B. Wolff, E. Lorenz, and O. Kramer. Statistical Learning for Short-Term Photovoltaic Power Predictions. In J. Lässig, K. Kersting, and K. Morik, editors, *Computational Sustainability*, volume 9570 of *LNCS*. Springer, Berlin, Heidelberg, 2016.

[109] E. Woods. Smart Street Lights Face Financial Hurdles. https://www.navigantresearch.com/blog/smart-street-lights-face-financial-hurdles, 2012. [Online; accessed 2016-04-04].

[110] L. Xu, W. He, and S. Li. Internet of Things in Industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, 2014.

[111] H. Yunhong, F. Liang, and H. Guoping. Privacy-Preserving SVM Classification on Vertically Partitioned Data without Secure Multi-party Computation. In *5th Int. Conf. on Natural Computation (ICNC)*, volume 1, pages 543–546, Aug. 2009.

[112] J. Zhang, D. Roy, S. Devadiga, and M. Zheng. Anomaly detection in MODIS land products via time series analysis. *Geo-spatial Information Science*, 10(1):44–50, 2007.

[113] Y. Zhang, J. Duchi, M. Jordan, and M. Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS) 26*, pages 2328–2336. Curran Associates, Inc., 2013.

[114] Y. Zhao, R. Schwartz, E. Salomons, A. Ostfeld, and H. Poor. New formulation and optimization methods for water sensor placement. *Environmental Modelling & Software*, 76:128–136, 2016.

[115] Y. Zheng, S. Rajasegarar, C. Leckie, and M. Palaniswami. Smart car parking: Temporal clustering and anomaly detection in urban car parking. In *IEEE 9th Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–6, 2014.

[116] K. Zhou and S. Yang. Understanding household energy consumption behavior: The contribution of energy big data analytics. *Renew. Sust. Energ. Rev.*, 56:810–819, 2016.

[117] I. Žliobaitė, M. Pechenizkiy, and J. Gama. An Overview of Concept Drift Applications. In N. Japkowicz and J. Stefanowski, editors, *Big Data Analysis: New Algorithms for a New Society*, pages 91–114. Springer International Publishing, 2016.