Title page

Chapter title:

Utilization of Big Data in Energy Internet Infrastructure

Names of the authors:

Songpu Ai (✉), Chunming Rong, and Junwei Cao

Affiliation and address of the authors:

University of Stavanger, 4036 Stavanger, Norway
Tsinghua University, Beijing 100084, P. R. China

E-mail address of the corresponding author:

songpu.ai@uis.no, 0047 48308615
chunming.rong@uis.no,
jcao@tsinghua.edu.cn

Utilization of Big Data in Energy Internet Infrastructure

Songpu Ai[1], Chunming Rong[2], and Junwei Cao[3]

Abstract: With the maturation of technologies such as communication, sensors, and networks, very large amounts of data are generated that are available for processing and analysis. With the popularizing process of the Internet of things (IoT), available data resources will become even more plentiful and diversified in the near future. To provide feasible solutions in data science for the key features of the energy internet, such as energy interconnection and routing, a big data architecture could be utilized in the energy internet infrastructure to provide large-scale analysis of massive various types of data. In this chapter, the utilization of big data in the energy internet infrastructure is explored. A three-layer big data architecture for usage in the energy internet is presented. The characteristics of data utilized in the energy internet and the potential requirements of the energy internet for the big data architecture are studied. Then, analytics methods that could be executed in the energy internet big data infrastructure are introduced. Real-time and offline analyses, as two types of analysis modes for different requirements of application scenarios, are described. Several well-known open-source big data tools are discussed. In addition, the open challenges of utilizing big data in the energy internet are proposed.

Keywords: Energy Internet, Big Data Architecture, Big Data Analysis, Big Data Platform, Real-time Analysis, Offline Analysis

[1] Songpu Ai (✉), University of Stavanger, Norway, songpu.ai@uis.no

[2] Prof. Chunming Rong, University of Stavanger, Norway, chunming.rong@uis.no

[3] Prof. Junwei Cao, Tsinghua University, P. R. China, jcao@tsinghua.edu.cn

1. Introduction

Managing and analysing data have always been challenges in the energy industry across infrastructures with different scales and distributed locations. Electricity companies on the generation side, network side, and consumption side have always struggled to develop reliable approaches to capture and analyse information to support scientists and researchers in achieving a better understanding of their products, services and customers to offer advanced solutions and save cost. In the past forty years, collected data has been analysed utilizing database technologies. In recent years, however, with the maturation of technologies such as communication, sensors, and networks, a very large amount of data is generated that is available for processing and analysis. With the popularizing process of the Internet of things (IoT), available data resources will become even more plentiful and diversified in the near future.

With this incoming surge of data, the generated data have the following notable characteristics: *Volume, Velocity, Variety,* and *Variability*, which are the so-called "4 Vs" [49]. For the energy internet, specifically, the volume of generated data could be very large. Every device within the energy internet can generate logs with a regular interval or by events. Persons can also create records whenever required. Over time, the amount of data to be analysed is considerable. Additionally, the velocity of data created in the energy internet can be especially rapid. To provide real-time analysis and execute further processing later on, the generated data flow needs to be stored, analysed and visualized in a timely manner, which is the characteristic of velocity. In addition, data are generated from a variety of sources in- and outside of the energy internet with a variety of types, for instance, sensor readings, images, videos, ecommerce records, and social media streams. Each data source can be independently collected and analysed. In addition, utilizing multiple types of integrated heterogeneous data to explore their interaction and the insight between them to achieve the purpose of analysis is an interesting and required topic [64]. Simultaneously, the variability of the above three Vs, which refers to changes in data volume, velocity and variety over time or between different energy internet subnets, must be considered. The changes include data flow rate, format, quality, etc.

In addition, many additional Vs can be presented to summarize the characteristics of big data in the energy internet. For example, the *Veracity* [23] of each data resource, which refers to the noise, biases, missing and abnormality in the data or the unmeasurable certainty of truthfulness and trustworthiness of the data, is required to be considered; *Valence* [32] refers to the connectedness of big data. Data items always relate with each other directly or indirectly. To find novel relationships between diverse types of data, involving additional data into consideration for comprehensive analysis with the support of big data techniques is the work of researchers. Although we can keep listing some other Vs that are attributed to big data, we prefer to utilize the 4 Vs we discussed in the previous paragraph as the primary characteristics.

However, one special V, *Value*, that we should mention here is the fundamental motivation of the entire big data technology revolution [44] and is also an important reason that we should consider the utilization of big data in the establishment of the energy internet. Big data is important because we believe that there is a huge amount of untapped value among the collected data [27]. Even though the data are too large to process with existing tools, the data arrive too rapidly to store and index optimally by centralized database technologies, and the data are too heterogeneous to fit any rigid schema, there is still enough value to dedicate capital and time into the corresponding research and development to break through technical problems and achieve additional insight from the data.

Big data is not a new technique. It is an association of existing and novel technologies. It is a scalable architecture of efficient storage, manipulation, and analysis. Traditional data architectures are no longer efficient enough to operate data through the architectures with the characteristics of volume, velocity, variety and variability simultaneously [37].

Big data has attracted attention in both academia and industry. Many ongoing and achieved innovations are engaged in or on big data technologies that obtain considerable results [28]. In business, enterprises utilize big data technology to understand and forecast consumer behaviour from all kinds of data sources they could collect. The quality, price, and improvement of products or services rely on the results of big data analytics. Management departments are also able to utilize big data to optimize company operational efficiency and reduce personnel costs.

In health care, big data has been utilized to analyse and forecast patient condition and disease progression, for example, analysing and comparing pathogenic characteristics integrating with patient physique through a very large number of medical cases. A more precise and customized treatment suggestion can be given by a big data system to assist the doctor in diagnosis and to reduce the incidence of patients.

Social media is another great known example of applying big data. Through analysing user online behaviour, including but not limited to instant messages, published content, online social networking and sharing activities, platforms such as Facebook, Twitter and LinkedIn are able to understand user behaviour patterns and preferences to improve service quality and efficiency. For example, if LinkedIn is aware that Alice works on machine learning using big data analysis, the website background will then more likely tend to push news and advertisements related to machine learning to Alice. A notorious case of abuse of big data in social media is the Facebook scandal. Cambridge Analytica Ltd. (CA) acquired approximately 50 million Facebook users' personal data and utilized the analysis results in the 2016 US presidential election and UK Brexit referendum in a tendentious manner. Facebook lost users and popularity. In addition, CA has filed for bankruptcy. It is a warning for all researchers and companies that protecting the privacy of data is an important principle, including research and development in the energy field. Additionally, the General Data Protection Regulation (GDPR) (EU) became enforceable in May 2018. It gives all individuals within the EU control of their personal data.

China and the US also have similar legislation being developed. We remind the reader here to comply with local laws while conducting research and development.

The utilization of big data in the energy field shows that existing studies generally focus on local and regional solutions with single functionality, as surveyed in [36]. More existing studies have been conducted to establish smart grid or smart city solutions. Project and research objectives are mostly limited to the microgrid level. Research on large-scale, multi-functional big data platforms operating in the energy internet is rare. However, as we discussed above, the capability of performing large-scale analysis on massive amounts of data is the critical advantage of big data technology. To provide feasible solutions in data science for the key features of the energy internet, such as energy interconnection and routing, big data architecture, analysis methods and platform building in the energy internet are introduced and analysed in this chapter as a reference for energy internet researchers to better utilize big data as a powerful tool. In addition, the open challenges of utilizing big data in the energy internet are discussed. The remainder of this chapter is structured as follows. Sect. 2 introduces the architecture of big data specifically in the energy internet field. Big data analysis methods are summarized in Sect. 3. Sect. 4 provides an overview of big data platform building. The existing open challenges of utilizing big data in the energy field are presented in Sect. 5.

2. The Architecture of Big Data

If we focus on the technical level, big data in fact is an integrated term for a stack of technologies. The realization of big data requires an appropriate combination and cooperation among technologies from various disciplines. Relative technologies include data collection, storage, management, manipulation, analysis, results display, etc. Even though for each application scenario with specific conditions and requirements, a particular big data stack should be tailored for the implementation, a similar architecture is utilized in most of the implementations.

In this section, a layered reference architecture is presented to discuss the technologies of the big data stack for the energy internet, while the specific characteristics and requirements of the big data architecture targeting the energy internet scenario are presented.

2.1 The Architecture of Big Data

Big data is a flourishing technology stack. Novel components and functions are booming in the big data landscape, enriching and evolving the entire stack constantly. Diverse architectures are proposed and implemented in research and

industry. Industry giants such as Google, Microsoft, and IBM as well as academic leaders such as the University of Amsterdam and University of California, San Diego, all hold their own proposed architectures. Nevertheless, certain key components/common tasks are present as layer-like structures in the majority of architectures, which are data collection, the storage and management layer, the data analytics layer, and the application layer, as shown in Fig. 1.

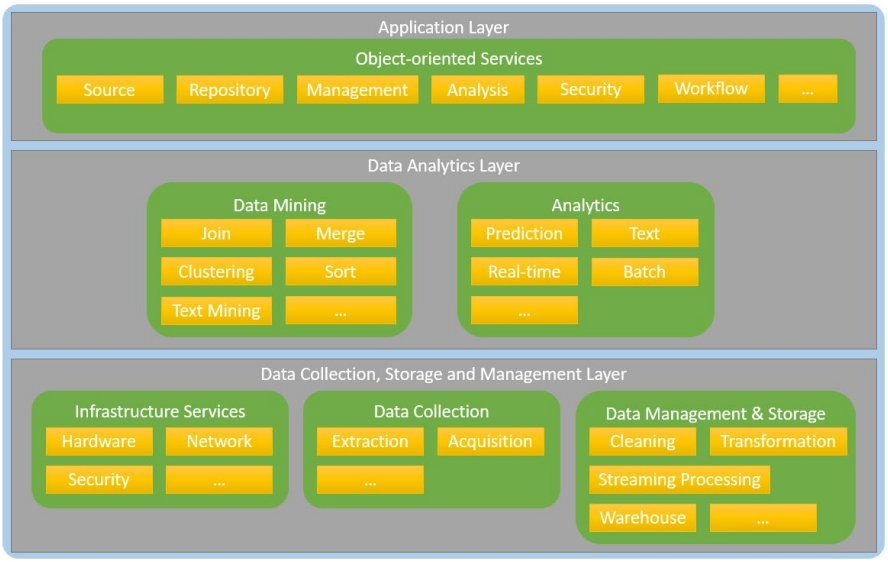

Figure 1 Layer-like structure of big data.

- Data Collection, Storage and Management Layer
  The data collection, storage and management layer is the basis of the entire big data architecture. As its name says, this layer first engages in collecting all types of data. The data can be structured, semi-structured or unstructured data records or data streaming from diverse data sources such as sensor readings, images, videos, ecommerce records, and social media platforms.
  Then, considering scalability, reliability, security, stability, and other reasons such as the data size and cost of communication, data are often stored in a distributed file system, such as the Hadoop Distributed File System (HDFS) [17] with duplications. The data can be stored as either SQL or NoSQL.
  Since the system is distributed, this layer also performs tasks to maintain the functionality of the entire system when node (a distributed server where data is stored) failure happens. Data should be able to be extracted, changed, and deleted by the upper layers while nodes leave and return.
  Above the storage system, a global resource manager (such as YARN [10]) is allocated to manage data usage and computational power.

To handle data with different formats, various data management tools are also engaged, such as Gephi [33] for graph data and MongoDB [45] for documents.
In the big data architecture, the data collection, storage and management layer offers proper data material for further utilization by the analysis and application layer. Several open-source project-based big data services for the data collection, storage and management layer, such as HDFS and YARN, are introduced in Sect. 4.

- Data Analytics Layer
  The data analytics layer is the intermediate portion of the architecture and is also the core of the entire stack. In this portion, data stored in the distributed file system can be operated and processed in real-time, near real-time or afterwards to provide diverse analytic results for applications. To realize data manipulations on data stored in numerous nodes simultaneously, parallel computing technology needs to be adopted [67].
  Since the big data stack is normally built on at least hundreds of nodes [51], traditional parallel techniques such as Open Multi-Processing [50] and Message Passing Interface [46] are no longer efficient enough to fit the novel implementation condition. Thus, the MapReduce [18] framework was developed to realize parallel processing on a large scale and with scalability.
  The fundamental idea behind MapReduce is "moving computation is cheaper than moving data" [17]. Therefore, MapReduce puts computational power and activity towards the data side and just obtains the results back, instead of transmitting data back and forth. By following this philosophy, the cost of communication and the stress on the computation centre can be reduced. However, MapReduce supports only one type of programming model, the map-reduce model, which is introduced in more detail in Sect. 4.2.4. With continuous technological advancement, more flexible, object-oriented processing frameworks supporting more parallel programming models have been developed and released for both batch and stream processing of data mining and analysis, for instance, Spark [8] and Flink [6]. More details about the well-known processing framework Spark are discussed in Sect. 4.2.4.
  By using batch and stream processing frameworks, big data analytical approaches such as deep learning can be achieved to study the association between variables and provide predictions/summaries to obtain the untapped value from big data. Several typical big data analytical approaches are presented in Sect. 3.
  The data analytics layer in the big data architecture handles the data analysis requirements from the application layer and submits deep analytic, predictive analytic, and/or summary analytical results for further utilization by top-layer applications.

- Application Layer
  The application layer is the topmost layer of the big data architecture. This layer offers object-oriented services.

There are many tools that have been developed to realize different specific services. For instance, Hive [11] is a data warehouse infrastructure offering ad hoc querying.

Moreover, we should note that there is no obligation that a tool should only belong to the data analytics layer or application layer. Depending on requirements, one tool, for example, Hive, can be either a portion of an integrated solution, as utilized in Facebook Messages [21], or an application itself, such as a data query platform [62], which is utilized by Facebook as well. Additionally, there is no restriction that a product/platform covers only one layer. As an example, the Spark [8] platform provides an in-memory big data processing framework for analyses and predictions as well as multiple practical tools that help it to be engaged in distinct types of applications. From this point of view, Spark covers both the intermediate and top layers. Furthermore, the Microsoft big data platform, Dryad [43], is capable of covering the development of an entire stack as utilized.

Moreover, tools can be adopted in combination to implement more complex functionalities. Furthermore, a stack should be able to run multiple services. Hence, a workflow management system is required in the big data stack to integrate, schedule, coordinate and/or monitor tools and services.

In addition, analysis results or extracted data items should be capable of interacting with other programs such as visualization tools (for instance, Tableau [61]) and decision support mechanisms at this layer to assist people in understanding and handling the situation well.

Through the implementation of the three-layer architecture discussed above in the energy internet, this approach could be able to realize an entire workflow from big data acquisition, management, and analysis to responses for a massive volume of various types of data that are generated within the energy internet or are captured from related resources with variability in data formats and generation velocities.

2.2 Implementing Big Data in the Energy Internet

In general, to build the energy internet through informatization and intellectualization to solve problems including improving equipment utilization, safety and reliability, power quality, and access to renewable energy, the introduction of big data analytics into the energy internet appears to be an indispensable technology roadmap at present [41]. Determining how to integrate the software and hardware requirements of the big data stack together with the requirements of the energy internet and existing energy and communication infrastructure to create an efficient

and affordable solution is a significant issue that energy internet practitioners need to solve jointly.

In this subsection, the characteristics of generated data in the energy internet and some potential requirements of the energy internet for big data architecture are discussed.

2.2.1 Characteristics of Energy Data

With the development of IoT technology, many devices in the energy internet are capable of generating and publishing data. Discussing and studying the characteristics of data generated in the energy internet is important for us to establish the big data stack and provide big data analyses and prediction services. From our perspective, data in the energy internet have the following characteristics:

- Data volume is very large and is generated with fast speed, including a large amount of streaming data.
  In the energy internet, each equipment can keep generating its status log, including but not limited to its *U, I, P,* and *Q* records. The generation frequency can be quick enough by selecting appropriate configurations. Hence, collecting, processing, analysing and giving responses through the big data stack with very large throughput is a challenge that needs to be solved, which is further discussed in Sect. 5.

- Considerable amounts of data generated in the energy internet are monitoring data, which are not critical for energy internet operations. However, once a situation fluctuates, the useful information is very dense.
  Data generated by equipment are mostly in a regular situation when the power prediction, scheduling and management procedures of the energy internet work well. Hence, in many circumstances, the data collected in the energy internet are used to confirm the prediction and management results and to monitor the situation. However, if an unexpected situation appears, the density of useful information in the data is high. Real-time responses need to be given through the analysis results derived from the data.

- The main portion of the data includes the real-time status information throughout the grid (such as *U, I, P,* and *Q*) as well as communication data about demand and supply. Additionally, there are various types of data generated outside the energy internet that should also be captured, stored and analysed by the energy internet big data stack, such as weather predictions and holiday pattern announcements.

2.2.2 Potential Requirements in Energy Internet Big Data Architecture

Combining the expectations of the energy internet to increase equipment utilization, promote safety and reliability, boost power quality, and advance renewable energy access, with the characteristics of energy internet data, we consider that the energy internet big data architecture should have the following requirements:

- Capability to process and monitor large-scale mass streaming data in real-time.
  To collect, process and manage data generated for the energy internet, the big data stack should be able to handle large-scale mass streaming data in real time. However, a certain amount of real-time data is not critical for energy internet operations and does not have a high information density for big data analyses. Therefore, the stack should be capable of "just" monitoring these data streams but not paying "too much" attention to them unless they are called by some applications (in real-time or as historical records).

- Capability to process and analyse in parallel large-scale nodes whenever necessary.
  When a demand or supply fluctuation appears, the energy internet big data stack should be capable of providing auto-decision making or decision support through processing and analysing the streaming data as well as historical data. If the fluctuation affects the load/supply scheduling in a wide area, parallel processing and analysis of the large-scale streaming data is demanded in the energy internet big data architecture.

- Low-latency real-time feedback, response, and decision making/support.
  The requirement of energy internet big data analysis on timeliness is sensitive. Real-time treatments, such as transient state balancing, are always important for power quality, grid stability, etc. In addition, response and decision making/support here do not refer to a centralized mechanism but to a distributed decision making/support mechanism or framework. By implementing treatments at different locations and on various infrastructures simultaneously, even lower latency is expected to be achieved.

- Scalability, stability and robustness in data analysis and prediction.
  The energy internet big data stack should be scalable along with the development of urban/rural areas, access to renewable energy, etc. In addition, the analysis and prediction should be robust and stable during issues such as network failure and grid damage. Furthermore, the stability and robustness of the stack are also crucial since the services that the stack offers relate to the most important terminal energy in modern society—electricity.

- Security of data, communication, and decision making, as well as protection of user privacy.

Historical energy data are required to be stored and manipulated safely. Similarly, the corresponding communication and decision making both need to be performed with security guarantees. In addition, energy data are relevant to privacy, not only for individuals but also for enterprises. Thus, the protection of user privacy is also needed for the big data stack.

3. Big Data Analytics Methods

In the energy internet field, massive various types of data are collected and required to be analysed [36]. With the development and release of new generations of big data processing frameworks and service platforms, an increasing number of database transformation operations, such as add, join, and filter, are supported. Most traditional analytical methods are able to be implemented in a big data stack easily and efficiently. There are tools or tool combinations that can provide similar data operations, interfaces and language environments as database management systems for a big data stack, such as Apache Impala [14], to make their users easily switch from a database to a big data stack.

The analytical capability of a big data stack is much greater than this. Along with the iterative operations that are supported by many analytical frameworks, machine learning as an intense user of iterative operations is widely utilized in big data analysis. Analytical applications include regression, classification, clustering, etc. Regarding the type of learning, the tasks can be classified into two main categories: supervised learning and unsupervised learning.

In this section, the two categories of big data analysis methods are introduced. Several commonly used analysis methods in the two categories are discussed. Moreover, deep learning and ensemble learning are presented as two notable methods that have promising potential in the energy internet field.

3.1 Supervised Learning

Supervised learning is a category of machine learning that learns the mapping between an input data set and the output data set (target). The objective of supervised learning is to precisely forecast the target for a given input. Frequently utilized supervised learning models include regression, random forest (RF), adaptive boosting (AdaBoost), Naive Bayes, artificial neural networks (ANNs), k-nearest neighbours (KNN), support vector machine (SVM), etc.

Regression analysis, including linear regression, logistic regression, etc., is used to find the most likely mathematical explanation between the dependent variable and independent variables. The advantage of this approach is that after the learning process, we obtain the mathematical relationship of the problem. By checking the residuals, the accuracy of the mathematical model can be verified. However, regression analysis needs researchers to assert the type of mathematical expression manually, which is an empirical task. More details on regression analysis can be found in [65].

RFs use decision trees as base prediction models to classify input data with different labels [22]. A decision tree is a tree-like graph used to model the different classifications among sets of input features. An RF first trains multiple individual decision trees constructed with a certain number of randomly selected features. In these trees, leaves correspond to the target classes, and branches represent the different variables of features selected by the tree that lead to those class labels. To classify a test input vector, it is passed through the trees in the forest. The forecast results are voted on by all trees with their own decisions.

Adaptive boosting (AdaBoost) is another type of learning method that uses a sequence of decision trees as base prediction models to produce a forward stagewise additive model. In AdaBoost, an additional weight feature is allocated to each training sample [57]. The weight of a sample is a response to the importance of forecasting the particular sample correctly. In each boosting iteration, according to the forecast result, a vote coefficient of the iteration is calculated, and the weight of each sample is updated. Incorrect training samples get higher weights to receive more attention in the next iteration of operation. A better iteration yields a higher vote coefficient in the final vote. More details on AdaBoost can be found in [24].

A Naive Bayes algorithm is an efficient probabilistic classifier that applies Bayes' probability theorem with the assumption that the input features are independent of each other [38]. According to the distribution of input features (for instance, Gaussian or multinomial), disparate Naive Bayes classifiers have been developed. More details on Naïve Bayes can be found in [47].

Support vector machine (SVM) is a machine learning model that attempts to find the optimal hyperplane to separate a dataset into two groups [60]. The direction of optimization is to find the hyperplane that has the maximum margins with the two groups. Margin here is the distance between the hyperplane and the data points (one or multiple) that are the closest to the hyperplane. The so-called support vector refers to the vector from the hyperplane directed towards the closest points. SVM was adopted in [52] to forecast the energy market and utilized in [26] to assess the solar radiation of a day to assist renewable energy generation.

K-nearest neighbours (KNN) is a widely used instance-based classification algorithm. It provides class forecasting based on the "distance" relationship between a testing sample and training samples. The "distance" can be decided by any distance functions (e.g., Manhattan, Euclidean, Minkowski, and Hamming) [63]. The classification is computed through a simple/weighted vote within a certain number of nearest neighbours of the testing sample. It was utilized in [1] as a component algorithm in an ensemble learning method to recognize the activities of ageing people in smart homes.

Artificial neural networks (ANNs) are widely utilized in smart home solutions, such as electrical appliance power profile classification in residential buildings [59] and human activity recognition in smart homes [20]. These networks model a biological neural network as an interconnected group of nodes called neurons. In each neuron, the weighted summation of an input vector is sent to the activation function to attain the output of the node. Normally, nodes in the network are structured as feedforward layers. Feedforward here refers to the fact that the outputs of these layer nodes are used as the inputs of the next layer. The data flow from the first (input) layer to the last (output) layer without looping back. When an input row is processed through the network by edges, a network output is obtained. An ANN learns by updating the neurons' weights after each training iteration. The weights are updated in the direction of minimizing the cost function of the problem.

3.2 Unsupervised Learning

Unsupervised learning uses unlabelled data as input to let the machine study the "structure" of the data. Unsupervised learning is normally utilized on problems that do not have a determined solution, which is also the reason why the data are unlabelled. Hence, it is hard to evaluate the results, which is an important difference between unsupervised and supervised learning. Commonly adopted unsupervised learning approaches include k-means clustering and ANNs. Within the energy internet big data stack, unsupervised learning could be useful for tasks such as auto-classifying sudden events to learn manual treatments to provide auto-decision making and decision support services.

K-means clustering uses the distances between each data item and the cluster centroids in the dataset vector space to classify data items into a certain number of clusters. The number of clusters should be assigned manually. The initial centroids are randomly selected from the vectors where the data items are held. The clusters are updated after each iteration of study by renewing the centroids of the clusters. The learning process finishes when the centroids no longer change or the changes remain within a certain range.

The learning process of an ANN in unsupervised learning is similar to that in supervised learning. The cost function is decided depending on the task, for example, the objective of unsupervised learning, as well as a priori assumptions such as the implicit properties of the model and observed variables.

3.3 Deep Learning

Deep learning attempts to use a multi-layer structured learning model to study the data, which can be both supervised and unsupervised learning. Neural networks, as an important approach to structure multi-layer architectures, have been widely discussed and adopted in deep learning studies. Neural network-based deep learning architectures, such as deep neural networks, recurrent neural networks and convolutional deep neural networks, have become representative deep learning approaches.

In general, a deep neural network (DNN) refers to an ANN with multiple feed-forward hidden layers. Hidden layers are the intermediate layers between the input layer and output layer. However, in simple ANNs, only one hidden layer is involved. By introducing multiple hidden layers into the network, DNNs have the potential to model complex data with fewer neurons than single hidden layer networks [30].

A convolutional neural network (CNN) is another type of feed-forward neural network that consists of one or multiple convolutional layers, pooling layers, fully connected layers, etc. CNNs have achieved better performance in image recognition studies [42].

A recurrent neural network (RNN) connects neurons by a directed graph along with a sequence, which enables the so-called “memory” in the network. It is capable of exhibiting dynamic behaviour on time series. Long short-term memory (LSTM) [34] and gated recurrent unit (GRU) [29] are two types of complex recurrent units for RNNs to promote the advantage of memory by gated state [3].

3.4 Ensemble Learning

Ensemble learning works to integrate multiple learning methods to obtain better analytical performance rather than focusing on a specific algorithm [55]. It was

developed based on evaluation results of different machine learning models for historical data. A common realization of ensemble learning is taking a vote among multiple promising models to obtain the forecast result [54].

In addition, many other model integration architectures have been adopted in studies to adapt to realistic requirements, which is called hybrid ensemble learning [2]. Prediction of device status in smart homes was proposed by integrating random forest and gradient boosting as the basis of an ensemble method [19]. Using a hybrid ensemble learning model to predict time series data with a weighted mean of the forecast results of several algorithms was discussed in [66]. In [2], a two-layer hybrid stacking ensemble learning model was employed to forecast EV charging demand.

Moreover, one additional objective or benefit of utilizing ensemble learning could be to improve the stability of the analysis [3], which is particularly required by the predictions in the energy internet.

4. Big Data Platforms

This section provides an overview of big data platform building in the energy internet. Real-time and offline analysis platforms are discussed. Some well-known open-source big data tools that could be used in energy internet big data platforms are introduced as well.

4.1 Real-time and Offline Analysis

When we consider establishing a big data platform, or stack based on its structural characteristics, the three-layer architecture discussed in Sect. 2 or a similar architecture is often followed. There are numerous tools for each aspect of the stack. The tools provide us with diverse kinds of services with different features. When we choose from them to build our stack, speed and scale are the two essential aspects that we need to consider. Due to the requirements of different application scenarios, two types of analysis modes, real-time and offline analysis, should be available in the energy internet big data stack based on the trade-off between speed and scale.

4.1.1 Real-time Analysis

Real-time analysis, just as its name suggests, tends to access, process, analyse data and to give responses as quickly as possible [39]. It is normally utilized in situations where the situation constantly changes, immediate analyses are required, and the response should be executed with very short latency.

Implementations of real-time analysis mainly use two types of structures. One uses a traditional relational database in a parallel processing cluster, which is not capable enough to satisfy the growing requirements of speed and scale. The other structure is integrating in-memory analytics platforms with a distributed file system [40], which is very good for performance.

Existing tools include portions or the entire platform of Spark [8], Storm [9], Beam [4] from Apache, Greenplum [31] from EMC and HANA [56] from SAP, etc. In the energy internet, potential utilization areas include real-time electricity demand prediction, pricing adjustment, load auto-balancing/scheduling, etc.

4.1.2 Offline Analysis

In contrast to real-time analysis, offline analysis focuses on more comprehensive data processing and analysis. It is able to make use of larger amounts of and more complex data in more sophisticated analysis methods.

For offline analysis implementations, data are normally already acquired and stored in the stack in advance, or the incoming data rate and response requirement are not high. Then, more sophisticated and, at the same time, more time-consuming operations are able to be utilized during the data processing. For example, we can realize iterative operations with fewer restrictions in offline analysis, which is one of the fundamental necessities of employing deep learning.

It should be mentioned that to promote processing speed considerably, real-time analysis technologies as well as the frameworks that we mentioned in Sect. 4.4.1 could be adopted for offline analyses as well.

In summary, real-time analysis has novel requirements for big data analysis. Hence, techniques for instance in-memory processing are developed to fill the gap. However, currently, the gap has not been fully filled. We are in a situation in which some relatively simple operations have been successfully achieved while some complex operations have yet to be realized. Since well-known platforms such as Spark and Storm [9] have supported real-time analysis functionality at a certain level, we can utilize real-time analysis operations together with complex operations that still currently belong to offline analysis to find the sweet spot between speed and scale.

4.2 Open-source Big Data Tools

Some well-known open-source big data tools are introduced in this subsection. Along with big data workflows, namely, acquisition, storage, management, analysis and response, we introduce one or two tools for each workflow pivot.

4.2.1 Acquisition Tool: Kafka

Data should be acquired by a big data stack before any further manipulation. To improve data acquisition efficiency and reduce the cost of format conversion during usage, acquisition tools have been developed based on distributed file systems, such as HDFS.

Existing tools include Kafka [7] developed by LinkedIn and Apache, Chukwa [5] developed by Apache Hadoop, Scribe [58] from Facebook, TimeTunnel [35] from Taobao, etc. Here, we introduce Kafka as an example of an acquisition tool.

Kafka is a high-throughput, low-latency, fault-tolerant stream processing platform that subscribes streaming data and distributed republishes the data in a proper way to feed the storage portion or for other purposes. It is able to collect and transmit hundreds of MB of data in each second.

In the Kafka architecture, it acts as an intermedia role between data generators/publishers and data consumers/subscribers. Data records (messages) are pushed to Kafka as key-value pairs. Kafka runs one or multiple servers (brokers) to receive the messages. The pairs are grouped by keys (topics) in each broker. Consumers can poll brokers to obtain messages from Kafka, as shown in Fig. 2. As the key components, the brokers store the published messages and prepare them plainly for consumers to pull their required data at the rate they prefer.

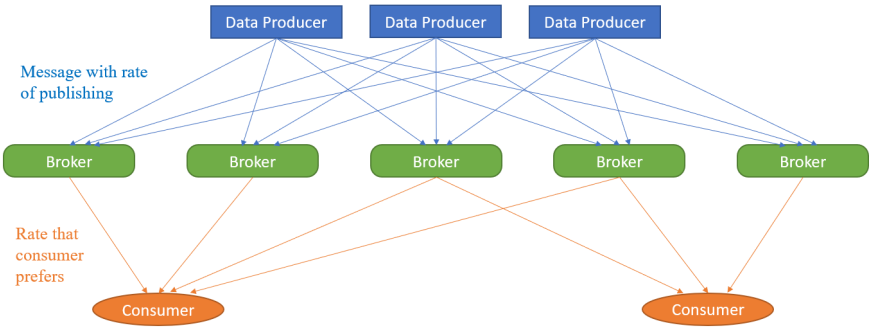

Figure 2 Architecture of Kafka.

4.2.2 Storage Tool: Hadoop Distributed File System (HDFS)

The HDFS [17] is the most famous and widely used distributed file system. It uses one or multiple servers to store split large files in blocks.
The HDFS architecture is a master and slave structure that consists of a cluster of nodes. The communication between nodes is based on TCP/IP. Each node (both the master and slaves) has a DataNode to store blocks that are allocated by the master. The master contains an additional NameNode to manage the namespace of the file system and to regulate file access from clients of the system. A file is first split into blocks with a standard block size, which is decided by the NameNode. Then, the blocks are stored in a set of DataNodes, as described in Fig. 3. To promote reliability, each block is replicated a certain number of times, and the replications are stored in different nodes.

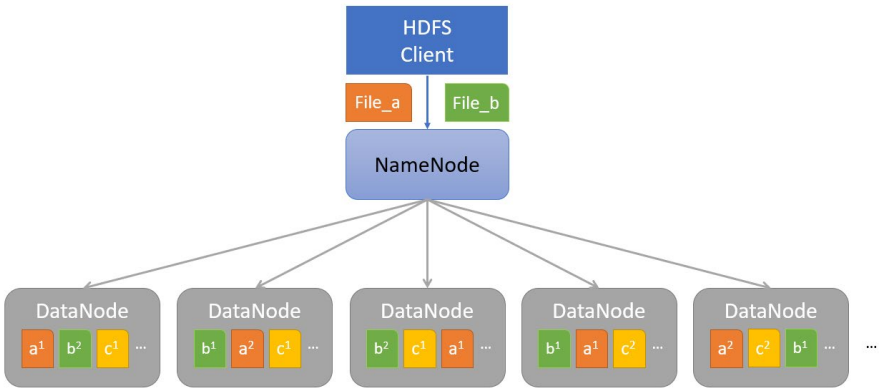

Figure 3 Data storage in HDFS.

4.2.3 Resource Management Tool: Yet Another Resource Negotiator (YARN)

YARN [10] is a global resource manager on top of the HDFS used to schedule and optimize cluster utilization. Optimization can be practised in different criteria, such as capacity and fairness.

In the architecture of YARN, there is a ResourceManager (RM) to arbitrate the resource usage (CPU, memory, disk, network, etc.) among all applications running on the file system. Each application (one or multiple jobs) is assigned a master manager to negotiate resources from the ResourceManager and execute the application. Moreover, each node is assigned a NodeManager to report the resource and resource usage regularly to the ResourceManager. A brief overview of how an RM obtains information from other YARN key components is presented in Fig. 4.

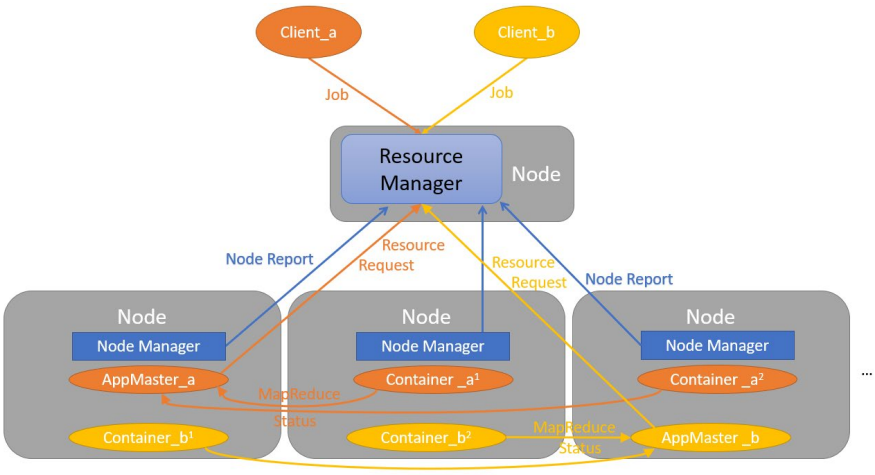

Figure 4 Operations of an RM obtaining information within YARN.

YARN supports many types of processing models operating on different execution engines, such as Tez [15], Spark [8], Flink[6], and Dryad[43], as well as big data applications, such as HBase [13].

4.2.4 Analytics Tool: MapReduce, Spark

To analyse massive data stored in a distributed system, a parallel programming framework is required to achieve efficient data processing and computation. Hadoop MapReduce [18] is an open-source implementation used to execute the MapReduce processing model on the HDFS.

The processing model of MapReduce only has two sequential stages, i.e., map and reduce. In the first stage, a map program is mapped towards certain nodes that store the blocks of input data. The map program is executed within the nodes in parallel. Then, in the second stage, the executed intermediate results are shuttled towards a small number of nodes to merge the intermediate results as a smaller set. Through several rounds of merging, the final output is achieved. The process of merging intermediate results towards the output is called “reduce”. The entire procedure is illustrated for a word count process example in Fig. 5.

MapReduce is a simple but powerful processing model. The open-source MapReduce released by Apache, Hadoop MapReduce, simplifies the programming procedure of research and development on big data. Only two programs are needed to develop a MapReduce processing application.

However, Hadoop MapReduce still has several shortcomings as an early big data execution engine. A fatal shortcoming is that Hadoop MapReduce supports only one processing model, and no other DAGs or iterations are supported.

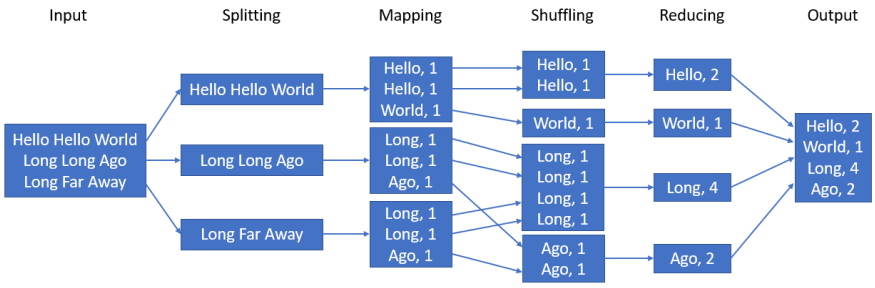

Figure 5 MapReduce process of a word count program.

Spark [8] is an open-source unified analytics platform first developed by AMPLab at the University of California, Berkeley, to support iterative and interactive big data processing pipelines. In addition to map and reduce operations, Spark supports a range of transformation operations, such as add, join, and filter. Moreover, there is no sequence restriction on pipeline implementation. Hence, Spark is capable of supporting high-performance iteration processing.

A resilient distributed dataset (RDD) is utilized to enable in-memory computation in Spark. An RDD is a read-only distributed set of data items. A parallel operation is not able to change an RDD but can build a new one to store the intermediate/final result set.

Spark includes multiple practical tools that help it to be engaged in distinct types of applications, such as streaming data processing and graph-parallel computation.

4.2.5 Application Tool: Hive

Tools for the application layer are mostly object-oriented to realize certain specific services. For distinct scenarios, various tools or tool combinations can be adopted to achieve a solution. Here, we use Apache Hive and Zookeeper as examples to provide the reader a preliminary impression of application layer tools.

Hive [11] was initially developed by Facebook and joined Apache later for big data querying. It is a data warehouse infrastructure providing data summarization and ad hoc querying. Users can use SQL-like language (HiveQL) to access data stored in various databases with different storage types efficiently. Hive or its further developed commercial versions are utilized by companies such as Netflix and Amazon as part of their big data implementations.

Apache ZooKeeper [16] provides centralized services to coordinate distributed big data stacks. It stores configuration values hierarchically and provides distributed stack group services, such as maintaining and synchronizing configurations and naming registries among nodes. This kind of maintenance and synchronization service is necessary for large distributed systems to enable highly reliable distributed

coordination. Zookeeper is utilized by companies such as Yahoo!, eBay, and Reddit.

5. Open Challenges for Utilizing Big Data in the Energy Internet

Although big data analysis has broad prospects in the energy internet field, several open challenges are required to be overcome ahead of utilizing it beyond demo cases. In this section, we introduce the main open challenges in implementing big data analysis in the energy internet.

5.1 Data Throughput

In the practice of the energy industry, a very large amount of data can be generated from infrastructure allocated at various places with a fast speed. For instance, records of energy receiving and forwarding between energy routers are created from each energy router continuously. The records can change quickly depending on situations such as transient state change. This method is required to handle the situation through processing, analysing the records and providing suggestions as soon as possible. Moreover, to obtain a big data stack with better performance, it is necessary to collect, store, manage, process, and analyse as much data as the system can, as rapidly as the system can. Hence, increasing the data throughput of the entire big data procedure, including data collection, storage, management, and analysis, is a continuing open challenge. This challenge remains along with the development of related technologies, which are able to generate and transmit more data with faster rates, such as the IoT.

One possible way to tackle the challenge stands with the IoT as well. Fog computing and edge computing can push the data processing even further towards the sensor itself or at the local sensor network level to reduce the cost of data transmission and central computing to promote the throughput of the big data stack.

In addition, integrating the big data stack with the energy internet is another aspect of this challenge because the processing requirements for the energy internet functionalities overlap with the big data services. If we design the physical, transport, and application layers of the energy internet with consideration of big data analytical requirements, then we are able to achieve a better solution for utilizing big data in energy internet circumstances.

5.2 Data Privacy and Security

For big data analytics, obtaining as much data as possible is a perpetual tendency. This is because a better performance analysis is more likely to be obtained by using enough data for analysis. However, in practice, it is not easy to obtain enough data to feed a big data stack. For researchers, electricity generation and consumption companies do not desire to let streaming data or huge amounts of records be taken outside their companies. It is even more difficult when negotiating with residential users to share their usage data. The situation has become worse after the Facebook scandal. Companies and individuals have become more conservative on data privacy and security issues. For companies in the power supply chain, obtaining data from upstream and downstream companies is also very difficult. Therefore, providing an efficient approach to obtain data with guaranteed privacy and security protection is an open challenge for both the research and industrial communities.

The enforcement of the GDPR can be a positive signal of normative data usage related to business. The GDPR regulates business behaviour around data, especially on data security and privacy protection. The severe sanctions can, to some extent, curb the occurrence of data abuse like that in the case of Cambridge Analytica Ltd.

Technically, blockchain [53] is a promising mechanism to protect data privacy and security. The data owner, which can be a company or an individual, can publish the authorizations of utilizing the data on the chain. An authorization includes the data abstract as well as the personnel of access, location of access, times of access, etc. A data user must obtain an authorization to access the data. Then, the utilization of data is easier to monitor by the owner and public. Thus, data privacy and security are able to be protected more efficiently.

5.3 Data Storage

At present, large amounts of data are generated with fast speed, but the progress of data storage capacity is not able to follow the increasing requirements on storage. The requirements generally regard two aspects, scale and speed, which are also the essentials of big data analysis.

For the challenge of increasing the speed of storage, in-memory databases [48] can be a potential approach to tackle this challenge. Through storing in-memory data, performance such as the reading and writing speed is faster than that for data stored on disks or on flash drives. However, the disadvantages of utilizing in-memory technology are still obvious at present: data should fit in-memory processing; in-memory databases have difficulty remaining persistent for long periods; databases should be loaded from/to disk images before/after usage; and data

communication between in-memory databases and other databases is not straightforward.

For the challenge of scale, web giants such as Google and Facebook keep building big data stacks with a larger scale of nodes for specific utilizations by themselves. However, the increasing number inspires our expectation of a large data storage scale in the energy internet. Public cloud service providers, such as Amazon and Microsoft, are also developing cloud storage-based big data stacks continuously for usage by industrial users.

5.4 Data Stream Processing

Although data acquisition frameworks such as Kafka and Flume [12] are capable of feeding the analytics layer hundreds of MB of streaming data per second, processing all valuable data in time, in other words, the timeliness of large-scale real-time processing, is still a challenge. For instance, despite the fact that Spark supports both in-memory iterative processing and streaming data applications, its technical implementation uses micro-batching to handle streaming, which allows stream granulation. It is not native stream processing, and the latency can be on the order of seconds with the configuration of the size of a micro-batch.

However, timeliness is a crucial requirement for utilization in the energy internet, including consumption pattern recognition, real-time power adjustment, scheduling, management, etc. Moreover, historical data should be capable of being extracted from the file system and interacting with new incoming data.

To overcome this challenge, big data processing frameworks, such as Apache Flink [6] and Beam [4] have been natively developed for streaming processing. These streaming processing frameworks just passed through their preview versions and entered the public view in 2018. The development of these frameworks could provide a promising solution for the timeliness challenge of large-scale real-time processing.

5.5 Data Opening

Data opening is a common challenge that researchers and public service providers face in many fields. For the purpose of data reuse and the social good, anonymous data should be stored for future usage. Data records and streams are commonly difficult to catch but easily disappear and are always valuable for research as well as public product development. These features are significant in energy data records.

To open up data records and data streams, financial support is necessary. Data with better quality, for example, more accurate, complete, and consistent data, often means more investment. In addition, better data quality also brings more accurate analysis, prediction and management. The trade-off regarding data quality is a challenge of data opening.

Furthermore, managing the opened data is also a challenge. Before data are opened, anonymization should be performed. However, determining the edge of privacy is difficult. Several levels of anonymization exist. If we anonymize all possible factors, the value of the data will also be lost. Another point that can be considered together with anonymization is who can access the open data. In general, if the data owner wants to check the data, it is not necessary to be anonymized. If it is a person or company without any credit background and confidentiality technology certification, he or it should only be able to access the data after the strictest anonymization process. Blockchain is also a potential solution to this challenge [25].

Summary

In this chapter, the utilization of big data in the energy internet infrastructure is explored. A three-layer big data architecture of usage in the energy internet is presented. The characteristics of data utilized in the energy internet and the potential requirements of the energy internet for the big data architecture are studied. Then, analytics methods that could be executed in the energy internet big data infrastructure are introduced. Real-time and offline analyses, as two types of analysis modes for different requirements of application scenarios, are described. Several well-known open-source big data tools are discussed. In addition, the open challenges of utilizing big data in the energy internet are proposed.

References

1. Agarwal B, Chakravorty A, Wiktorski T, Rong C Enrichment of Machine Learning Based Activity Classification in Smart Homes Using Ensemble Learning. In: 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC), 6-9 Dec. 2016. pp 196-201
2. Ai S, Chakravorty A, Rong C Household EV Charging Demand Prediction using Machine and Ensemble Learning. In: 2018 IEEE International Conference on Energy Internet (ICEI), 21-25 May 2018. pp 163-168
3. Ai S, Chakravorty A, Rong C (2019) Household Power Demand Prediction Using Evolutionary Ensemble Neural Network Pool with Multiple Network Structures. Sensors 19 (3)
4. Apache (2018) Beam. https://beam.apache.org/.
5. Apache (2018) Chukwa http://chukwa.apache.org/.
6. Apache (2018) Flink. http://flink.apache.org/.

7. Apache (2018) Kafka https://kafka.apache.org/
8. Apache (2018) Spark. http://spark.apache.org/.
9. Apache (2018) Storm. http://storm.apache.org/.
10. Apache (2019) Apache Hadoop YARN. https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html.
11. Apache (2019) Apache HIVE. https://hive.apache.org.
12. Apache (2019) Flume http://flume.apache.org/.
13. Apache (2019) HBase. http://hbase.apache.org/.
14. Apache (2019) Impala. https://impala.apache.org/.
15. Apache (2019) Tez. https://tez.apache.org/.
16. Apache (2019) ZooKeeper. https://zookeeper.apache.org/.
17. Apache Hadoop (2019) HDFS Architecture Guide. https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.
18. Apache Hadoop (2019) MapReduce Tutorial. https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html.
19. Bhole M, Phull K, Jose A, Lakkundi V Delivering Analytics Services for Smart Homes. In: 2015 IEEE Conference on Wireless Sensors (ICWiSe), 24-26 Aug. 2015. pp 28-33
20. Bier T, Abdeslam DO, Merckle J, Benyoucef D Smart Meter Systems Detection & Classification using Artificial Neural Networks. In: IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society, 2012. IEEE, pp 3324-3329
21. Borthakur D, Gray J, Sarma JS, Muthukkaruppan K, Spiegelberg N, Kuang H, Ranganathan K, Molkov D, Menon A, Rash S, Schmidt R, Aiyer A Apache Hadoop Goes Realtime at Facebook. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, Athens, Greece, 2011. ACM, pp 1071-1080
22. Breiman L (2001) Random Forests. Machine Learning 45 (1):5-32
23. Bureva V, Popov S, Sotirova E, Atanassov KT Generalized Net of MapReduce Computational Model. In, Cham, 2018. Uncertainty and Imprecision in Decision Making and Decision Support: Cross-Fertilization, New Models and Applications. Springer International Publishing, pp 305-315
24. Cao Y, Miao Q-G, Liu J-C, Gao L (2013) Advance and Prospects of AdaBoost Algorithm. Acta Automatica Sinica 39 (6):745-758
25. Chakravorty A, Rong C Ushare: User Controlled Social Media Based on Blockchain. In: Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, Beppu, Japan, 2017. ACM, 3022325, pp 1-6
26. Chen J-L, Li G-S, Wu S-J (2013) Assessing the Potential of Support Vector Machine for Estimating Daily Solar Radiation using Sunshine Duration. Energy conversion and management 75:311-318
27. Chen J, Chen Y, Du X, Li C, Lu J, Zhao S, Zhou X (2013) Big Data Challenge: A Data Management Perspective. Frontiers of Computer Science 7 (2):157-164
28. Chen M, Mao S, Liu Y (2014) Big Data: A Survey. Mobile Networks and Applications 19 (2):171-209
29. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv preprint arXiv:14061078
30. Deng L, Yu D (2014) Deep Learning: Methods and Applications. Foundations and Trends® in Signal Processing 7 (3–4):197-387
31. EMC (2018) Greenplum Database. https://greenplum.org/.
32. Fiore-Gartland B, Neff G (2015) Communication, Mediation, and the Expectations of Data: Data Valences Across Health and Wellness Communities. International Journal of Communication 9:1466-1484
33. Graph (2019) The Open Graph Viz Platform. https://gephi.org/.
34. Hochreiter S, Schmidhuber J (1997) Long Short-term Memory. Neural computation 9 (8):1735-1780

35. Huolong (2018) TimeTunnel. http://code.taobao.org/p/TimeTunnel/wiki/index/.
36. Jiang H, Wang K, Wang Y, Gao M, Zhang Y (2016) Energy Big Data: A Survey. IEEE Access 4:3844-3861
37. Katal A, Wazid M, Goudar R Big Data: Issues, Challenges, Tools and Good Practices. In: Contemporary Computing (IC3), 2013 Sixth International Conference on, 2013. IEEE, pp 404-409
38. Kelleher JD, Namee BM, D'Arcy A (2015) Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. The MIT Press, London
39. Laplante PA (2004) Real-time Systems Design and Analysis. Wiley, New York,
40. Lee J, Kwon YS, Färber F, Muehle M, Lee C, Bensberg C, Lee JY, Lee AH, Lehner W SAP HANA Distributed In-Memory Database System: Transaction, Session, and Metadata Management. In: 2013 IEEE 29th International Conference on Data Engineering (ICDE), 8-12 April 2013. pp 1165-1173
41. Liu S, Zhang D, ZHU C, LI W, LU W, ZHANG M (2016) A View on Big Data in Energy Internet. Automation of Electric Power Systems 40 (8):14-21
42. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE (2017) A Survey of Deep Neural Network Architectures and Their Applications. Neurocomputing 234:11-26
43. Microsoft (2019) Dryad. https://www.microsoft.com/en-us/research/project/dryad/.
44. Miller HG, Mork P (2013) From Data to Decisions: A Value Chain for Big Data. IT Professional 15 (1):57-59
45. MongoDB (2019) MongoDB https://www.mongodb.com/.
46. MPI Forum (2018) MPI Forum. https://www.mpi-forum.org/.
47. Murphy KP (2006) Naive bayes classifiers. University of British Columbia:1-8
48. Najajreh J, Khamayseh F Contemporary Improvements of In-Memory Databases: A Survey. In: 2017 8th International Conference on Information Technology (ICIT), 17-18 May 2017. pp 559-567
49. NIST Big Data Public Working Group (NBD-PWG) (2015) The NIST Big Data Interoperability Framework: Volume 1, Definitions. USA Patent,
50. Openmp (2019) Openmp. https://www.openmp.org/.
51. Oussous A, Benjelloun F-Z, Ait Lahcen A, Belfkih S (2018) Big Data Technologies: A Survey. Journal of King Saud University - Computer and Information Sciences 30 (4):431-448
52. Papadimitriou T, Gogas P, Stathakis E (2014) Forecasting Energy Markets using Support Vector Machines. Energy Economics 44:135-142
53. Pilkington M (2016) Blockchain Technology: Principles and Applications. In: Research handbook on digital transformations. Edward Elgar, UK, pp 225-253
54. Polikar R (2006) Ensemble based Systems in Decision Making. IEEE Circuits and Systems Magazine 6 (3):21-45
55. Rokach L (2010) Ensemble-based Classifiers. Artificial Intelligence Review 33 (1):1-39
56. SAP (2018) SAP HANA. https://www.sap.com/sea/products/hana.html.
57. Schapire RE, Freund Y (2012) Boosting: Foundations and algorithms. MIT press, London
58. Scribe (2018) Scribe. https://www.scribesoft.com/.
59. Songpu A, Kolhe ML, Jiao L External Parameters Contribution in Domestic Load Forecasting using Neural Network. In: International Conference on Renewable Power Generation (RPG 2015), 17-18 Oct. 2015. pp 1-6
60. Suykens JAK, Vandewalle J (1999) Least Squares Support Vector Machine Classifiers. Neural Processing Letters 9 (3):293-300
61. Tableau (2019) Tableau. https://www.tableau.com/.
62. Thusoo A, Sarma JS, Jain N, Shao Z, Chakka P, Anthony S, Liu H, Wyckoff P, Murthy R (2009) Hive: A Warehousing Solution Over A Map-Reduce Framework. Proceedings of the VLDB Endowment 2 (2):1626-1629
63. Walters-Williams J, Li Y Comparative Study of Distance Functions for Nearest Neighbors. In: Elleithy K (ed) Advanced Techniques in Computing Sciences and Software Engineering, Dordrecht, 2010. Springer Netherlands, pp 79-84

64. Wang Y, Shi Q, Song H, Li Z, Chen X (2016) Multi-source Heterogeneous Data Integration Technology and Its Development. Data Management and Three-dimensional Visualization of Global Velocity Mode l Crust 20:133
65. Weisberg S (2005) Applied Linear Regression, vol 528. John Wiley & Sons, New Jersey
66. Wichard JD An Adaptive Forecasting Strategy with Hybrid Ensemble Models. In: Neural Networks (IJCNN), 2016 International Joint Conference on, 2016. IEEE, pp 1495-1498
67. Yang C, Huang Q, Li Z, Liu K, Hu F (2017) Big Data and Cloud Computing: Innovation Opportunities and Challenges. International Journal of Digital Earth 10 (1):13-53