Big Data Service Architecture: A Survey

Jin Wang[1,2], Yaqiong Yang[1], Tian Wang[3], R. Simon Sherratt[4], Jingyu Zhang[1]

[1] School of Computer &Communication Engineering, Changsha University of Science & Technology, China
[2] School of Information Science and Engineering, Fujian University of Technology, China
[3] College of Computer Science and Technology, Huaqiao University, China
[4] Department of Biomedical Engineering, the University of Reading, UK
jinwang@csust.edu.cn, yangyqst@163.com, cs_tianwang@163.com, sherratt@ieee.org, zhangzhang@csust.edu.cn

Abstract

As one of the main development directions in the information field, big data technology can be applied for data mining, data analysis and data sharing in the massive data, and it created huge economic benefits by using the potential value of data. Meanwhile, it can provide decision-making strategies for social and economic development. Big data service architecture is a new service economic model that takes data as a resource, and it loads and extracts the data collected from different data sources. This service architecture provides various customized data processing methods, data analysis and visualization services for service consumers. This paper first briefly introduces the general big data service architecture and the technical processing framework, which covered data collection and storage. Next, we discuss big data processing and analysis according to different service requirements, which can present valuable data for service consumers. Then, we introduce the detailed cloud computing service system based on big data, which provides high performance solutions for large-scale data storage, processing and analysis. Finally, we summarize some big data application scenarios over various fields.

Keywords: Big data, Data processing, Data analysis, Cloud service model, Big data applications

1 Introduction

As the concept of big data first appeared in the journal Nature, it is described as large-scale data that can not be presented, processed and analyzed using existing technologies, methods and theories [1]. Big data has the following four typical characteristics, i.e., Volume, Variety, Velocity, and Value [2]. The statistics show that the economic aggregate of global big data market has reached US$58.9 billion in 2017, with the 29.1% increment. By 2020, the global big data market will create more than 121.4 billion US dollars. It is urgent to develop technologies and platforms with better performance to compute, process and analyze the large-scale data [3-4]. Big data technology can improve social governance and production efficiency, and promote scientific research [5-6]. There are complex and challenging tasks that can not be dealt with by traditional reasoning and learning methods, requiring innovative techniques, algorithms and infrastructure. Therefore, to tackle the new challenges of big data technologies, we take a in-depth study of the current big data service architecture.

This paper is devoted to analyzing the current big data service architecture, which is composed of three main layers. In the data collecting and storage layer, data sources in big data services are needed to be collected by corresponding equipment, and then the data in “pre-processed” state will be stored and processed in a distributed file system or database system. In the data processing layer, different processing frameworks are adopted according to different forms of data. The in-depth analysis of big data is currently mainly based on large-scale machine learning technologies, which can deeply mine the potential value of data. Finally, visualization tools are used to present results to data service consumers. In the application layer, there are applications of big data technology over various fields. In addition, in big data-based cloud computing services, software and infrastructure built on cloud model (i.e., SaaS, PaaS, IaaS) are utilized to process big data. The big data service architecture is shown in Figure 1.

In the remaining sections of this paper, Section 2 introduces the infrastructure of big data service architecture, which involves the collecting and storage of massive data. Section 3 presents the introduction of big data processing and analysis technologies. Section 4 introduces the cloud computing service models based on big data, and the integration of cloud computing and big data technologies. In the last section, we summarize some practical application scenarios of big data services.

*Corresponding Author: Jingyu Zhang; E-mail: zhangzhang@csust.edu.cn

DOI: 10.3966/160792642020032102008

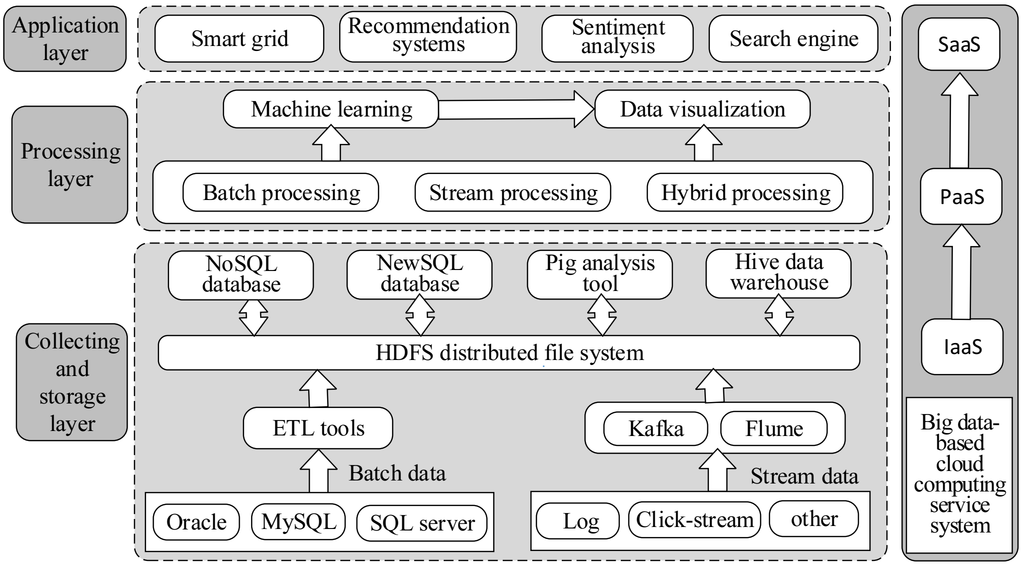

Figure 1. Big data service architecture

2 Data Collecting And Storage

In the era of big data, information integration usually needs to extract and load large amounts of data from massive data sources [7-8]. These distributed data are needing to be collected by appropriate equipment or software, and data storage management schemes should be provided for these massive data in the sequential processing steps.

2.1 Data Collecting

The forms of big data mainly include static batch data and dynamic stream data. Batch data is stored in a static form, and stream data is a continuous real-time data instance sequence. The streaming data will not be stored completely, and many elements will be discarded directly after processing [9].

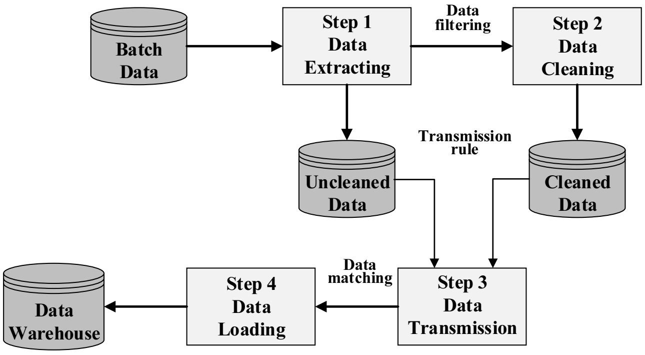

Figure 2. The process of ETL

Because of the instability of stream data transmission, streaming data collecting is different from traditional batch data collecting. For batch data from different data sources, ETL (Extract-Transform-Load) tools are usually used to realize the transmission and collection of various data types. ETL process is to extract data from the data sources, then transform and load it into the storage targets [10]. The process is shown in Figure 2. ETL removes corrupted or dirty data through data processing operations such as connection, transformation, and cleanup. The widely used ETL tools are Kettle, Datastage, Informatica, etc.

For the stream data that needs to be collected in real time, a collection tool that can guarantee the instantaneity, fault tolerance, stability and reliability is needed. Flume is a reliable and fault-tolerant distributed stream processing system that collects, aggregates and transfers a large number of log data from different sources to a centralized storage area [11]. Flume is usually built around the Hadoop ecosystem and it acts as a middleware between data sources and receivers [12]. And Kafka is a universal open source messaging system, which is mainly used to build real-time data pipelines and streaming applications [13]. In order to further optimize the control and processing speed of stream data, Kafka uses queues to process data, which can avoid the asynchronism of processing speed between data generation and processing [14]. The other famous systems include Facebook's Scribe, Taobao's TimeTunnel.

2.2 Data Storage

In the past decades, relational databases and structured data management techniques have been widely used [15]. According to the characteristics of big data, the storage systems adopted mainly include distributed file systems, NoSQL, NewSQL and other data management systems.

In 2003, Google developed a scalable file system for large-scale distributed storage called Google File System (GFS) [16]. It provides high aggregation performance of massive data, and meets the demand of users for large-capacity storage [17]. HDFS is a part of Apache Hadoop core project, and its design idea refers to GFS system [18-19]. HDFS at present is considered to be the most widely used big data tool, supporting redundancy, reliability, scalability for parallel distributed architecture systems [20].

NoSQL is a general term for a database whose data management mode is non-relational. This NoSQL data model involves key-value pairs, column families, graphs or documents. Table 1 describes several different types of NoSQL databases. For example, Because of the simple structure of the key-value pair model, it is not easy to cause data collisions and the programming model is easier to implement. Another data model is based on documents, using a key to identify each document. Unlike key-value storage, data in a document can be queried. Column family stores are inspired by Googles Bigtable [29], and its data is stored based on the column family model.

Table 1. Different types of NoSQL database

NoSQL databases	Types	Characteristics	Advantages	Disadvantages	Scenarios
Redis [21]	Key-Value	Key-value pairs; Simple data structure; High scalability; Low query update efficiency	Excellent read-write performance; Data persistence; Read-write separation	Without automatic fault tolerance and recovery function; Not supporting for on-line expansion	User information storage related to ID (key), i.e. sessions, configuration files, parameters
Memcached [22]			Memory-based data processing; Distributed and scalable mode; Fast processing speed	Small single cache data capacity; Not supporting for data persistence	
MongoDB [23]	Document-Oriented	Document-based data processing unit; Multiple data formats, i.e. JSON, XML	Fast access speed; Multiple data types; High query efficiency	Low read-write efficiency; Large space occupancy	Suitable for storing log files and analyzing data in real time
CouchDB [24]			Availability and concurrency; High flexibility	Low query efficiency; High maintenance difficulty	
Hbase [25]	Column-Family	Column-based data storage model; High data correlation in the same column family	Fast loading speed; Suitable for storage of large data; Efficient compression rate	Low multi-conditional query efficiency; Not directly supporting SQL statement query	Suitable for high concurrency operation on big data; Real-time read/write access
Cassandra [26]			Elastic scalability; Flexible schema; Multi-list data structures	Not supporting for ACID transactions and atomic operations	
Neo4j [27]	Graph-Oriented	Graph-based data storage formats, i.e. entities are vertices; High data correlation between entities; Efficient query performance	Efficient query performance; High performance graphics algorithm	Not supporting for large graph partition	Suitable for relational data storage, recommendation engines, community websites
GraphDB [28]			ACID transactions and atomic operations; Multi-model objects	High complexity	

NewSQL represents the new relational database that not only has the same scalablity as NoSQL, but still provides ACID and SQL services for transactions [30]. The current widely used NewSQL databases are: Spanner, MemSQL [31-32].

3 Data Processing

At the beginning of data processing, raw data should be cleaned, cropped and integrated in order to provide high performance big data services for customers. Table 2 introduces different data processing modes in the big data processing framework.

3.1 Data Processing

Because of the huge amount of data, the cost of data processing and analysis by accessing all the data is high, or even we can not complete the processing in a given time period. Hence, at the beginning of data processing, raw data should be cleaned, cropped and integrated in order to provide high performance big data services for customers. Table 3 summarizes the characteristics of the currently popular data processing frameworks.

3.1.1 Batch Data Processing

Since batch data is static and the amount of data is extremely large, data processing is usually performed using the distributed off-line computing processing method that is capable of parallel computing. In 2004, Google designed and developed MapReduce [33], which is a distributed popular programming model for processing big data sets. MapReduce allows users to build complex computations on big data sets without

Table 2. Data processing modes

	Date processing modes			
Items	Batch processing	Stream processing	Hybrid processing	Graph processing
Data characteristics	Large-scale; High accuracy	Continuous infinite real-time data sequence	Existing of both batch data and stream data	Relevant data formed by vertices and edges
Processing speed	Minutes level	Millisecond level	Millisecond level	Second level
System characteristics	Simple programming mode; Time-insensitivity; Data-intensive	Serialized, low latency, event-driven triggering	Diversified workloads; High fault tolerance; Low latency	Graph data with massive nodes and edges; High data correlation
Communication mechanism	RPC/HTTP	Message queue	Shared memory/broadcast	Message queue
Data storage	HDFS	Real-time input stream	Memory/disk	Distributed file systems/ databases
Typical processing frameworks	MapReduce	Storm, Samza	Spark, Flink	Pregel, Giraph
Application scenarios	Off-line analysis and processing of massive data, large-scale Web information search	Pure real-time analysis; real-time scheduling, continuous calculation	Iterative machine learning; Incremental calculation	Social network map; traffic road map analysis

Table 3. Characteristics of data processing frameworks

	Data Processing Frameworks							
Characteristics	MapReduce	Dryad	Storm	Samza	Spark	Flink	Pregel	PowerGraph
Scalability								
Real-time								
Reliability								
Data correlation								
Concurrency								
Data persistence								
Multilingual programming								
Memory processing								
Fault-tolerance								
Checkpoint mechanism								
Reprocessing mechanism								
Multi-mode operation								
Stream processing								
Micro-batch processing								
Batch computing								
Interactive query								

worrying about synchronization, fault tolerance, reliability or availability [34]. MapReduce usually divides the input data set into independent blocks, which are processed by parallel Map tasks. Then the output results of Map phase should be further processed by the following reduce tasks. In Reduce phase, shuffle processing distributes data partitions to Reduce nodes dynamically [35].

3.1.2 Stream Data Processing

The stream data processing pattern is suitable for processing data that requires a real-time response, therefore stream processing is needed as the data processing framework that can achieve low latency.

Storm is an open source distributed real-time stream data processing system [36]. Storm is similar to a real-time Hadoop computing system, which does not need to do complicated task scheduling, so it can reduce processing latency [37]. Storm has the following characteristics: (1) Simple programming model; (2) Scalability; (3) High reliability (unlike real-time systems such as S4 [38], Storm ensures that data can be processed completely); (4) High fault tolerance (Storm rearranges the problem processing units if some exception occurs during message processing) [39-40].

Apache Samza is a stream processing framework capable of efficiently processing large amounts of data [41]. The largest Samza implementation by LinkedIn can handle millions of messages per second during peak hours [42]. Meanwhile, the combination of Samza and Kafka can better take the advantages of the two frameworks. Kafka can provide fault tolerance, data buffering, state storage and other technologies for Samza, and the relationship is similar to the dependence of MapReduce engine on HDFS [43].

3.1.3 Hybrid Data Processing

Some tasks include both batch data processing and stream data processing. Many data processing frameworks support two types by combining the similar or related components and APIs, and can simplify different data processing processes.

Apache Spark is a new batch data processing framework with stream data processing capabilities [44]. Spark uses a method called Resilient Distributed Dataset. Because using RDDs to process data, Spark can speed up batch data processing by putting the entire process in memory [45]. Spark has better data processing performance than Hadoop without memory capacity limitations [46]. In addition, Spark Streaming component implements a method called Micro-batch that treats continuous flows as a series of micro-batch data, and handles these micro-batch jobs continuously [47]. Spark Steaming component has good fault tolerance and load balancing, but it still has insufficient performance compared to the complete stream processing frameworks.

Flink is an open source data analysis framework managed by Apache for batch and stream data processing [48]. Compared with other big data processing frameworks, Flink has its unique data processing methods. It is used with a persistent message queue (such as Kafka) to process data at different points of the persistent streams [49]. Although both Spark and Flink can handle hybrid data, the micro-batch processing architecture adopted by Spark will take more time for the data processing than Flink [50].

3.1.4 Graph Data Processing

In the massive data, some data called graph data are linked together in the form of graphs or networks. The number of vertices and edges of graphs has reached hundreds of millions. Traditional graph data computing frameworks can no longer meet the huge computing needs. At present, two kinds of graph processing frameworks are mainly used to deal with these large-scale graph data 51: one is a graph database capable of real-time data processing (e.g., Neo4j, OrientDB, and DEX); the other is a computing engine capable of parallel batch processing (e.g., Hama, Giraph and Regel).

Pregel is a popular batch synchronous parallel computing system launched by Google, and it introduces a vertex centered large-scale graph computing model 52. The Pregel framework is more efficient than MapReduce in dealing with iterative graph data computation. Pregel can provide a computing engine with excellent performance for the traversal, shortest path and PageRank computation for large graph data 53.

3.2 Data Analysis and Visualization

Big data analysis technologies (such as machine learning) are used to mine valuable data in the massive data, so as to support the prediction and analysis of future trends and patterns. Afterward, the information should be presented to data service consumers through data visualization. Figure 3 shows machine learning techniques based on big data.

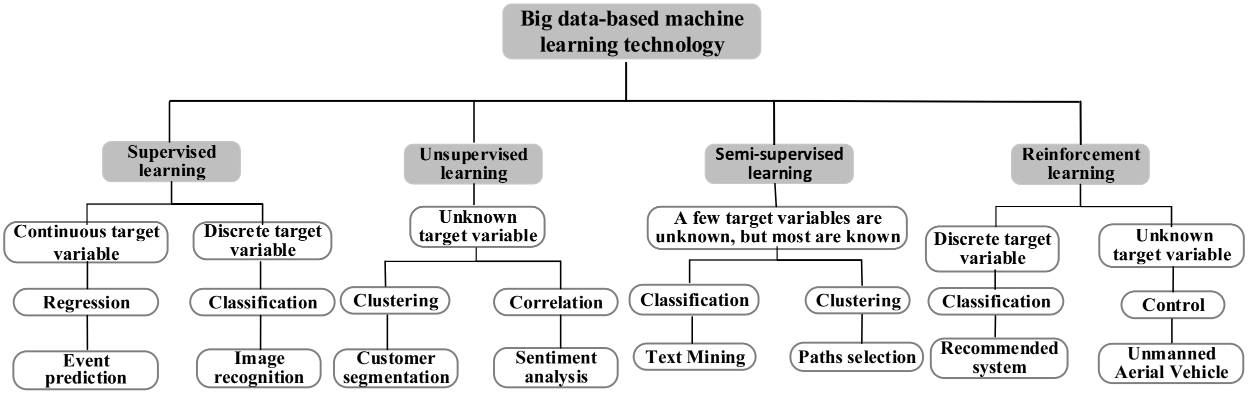

Figure 3. Big data-based machine learning technology

With the in-depth development of big data technologies, the machine learning is widely used for the in-depth big data analysis [54-56]. Machine learning is a research field focusing on theories, learning systems and algorithm attributes, which includes artificial intelligence, information theory, optimal control, cognitive science, mathematics and engineering, data mining, control systems, identification systems, informatics and so on [57-60].

Currently, machine learning has attracted more and more attention in terms of big data analysis. For the analysis of image data, Niu et al. [61] proposed a novel multi-scale depth model, which was used to extract rich and discriminative features that could represent various visual concepts. [62] proposed a novel unsupervised feature learning approach for mapping pixel reflectance to illumination invariant encodes. In [63], an efficient image retrieval method was proposed, which improve the overall retrieval rate. In the field of natural language processing, Liu et al. [64] proposed a feature selection method based on correlation analysis and Fisher, which can eliminate the redundant features. [65] adopted an unsupervised learning method to estimate the acoustic model of speech recognition, which does not need to transcribe the training data.

Data service consumers need to obtain valuable data after the processing and analysis. Data visualization usually uses tables and images to present data information for users. By vivid visual effects, data can be presented in a more understandable and intuitive form, thereby enhancing the attractiveness and persuasiveness. Through data visualization, data analysts use data trends, data patterns, relationships and other information to study the data more deeply from different dimensions, so as to further improve data analysis. Data visualization tools have been widely used, mainly including charting tools (D3, RawGraphs, Google Charts, etc.); map class forming tools(Modest Maps, Openheatmap, ColorBrewer, etc.); timeline forming tools (Cube, Timeflow, Dipity, etc.).

4 Big Data-based Cloud Computing Service Systems

The advantages of cloud computing technology lies in its powerful distributed processing engines, distributed databases, cloud storage and virtualization technologies. The establishment of cloud computing service system based on big data forms a excellent performance data cloud service platform. Figure 4 is the architecture of big data-based cloud computing service systems.

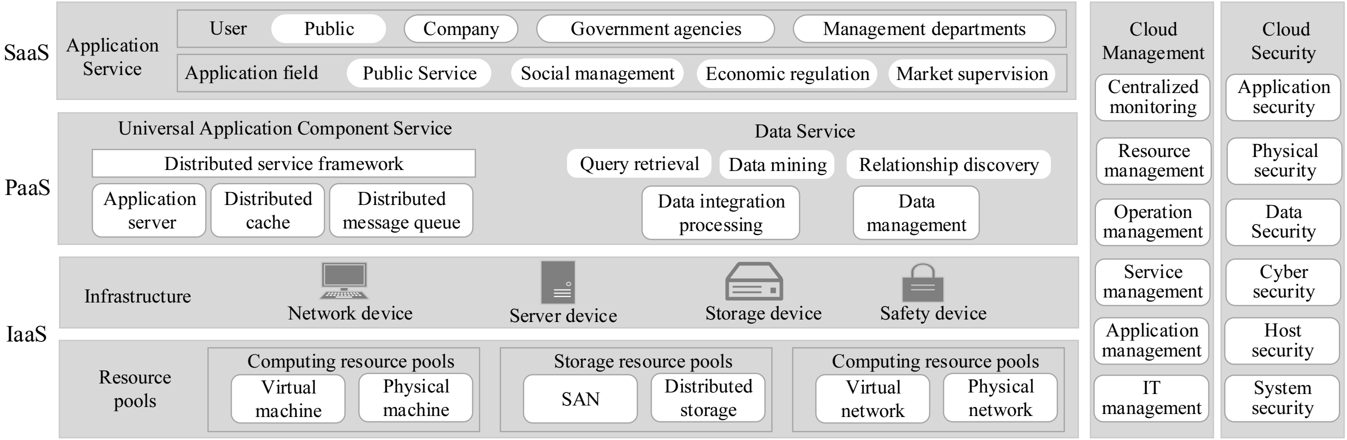

Figure 4. The architecture of big data-based cloud computing service

4.1 Big Data-based Cloud Computing Service Models

Cloud computing is a novel computing paradigm, which is used to implement a shared pool model of configurable computing resources (e.g., networks, servers, storages, applications and services), and it can be easy to quickly configure and publish cloud computing tasks [66-67]. Big data workloads often experience repeated changes of scope and size, and powerful processing schemes are often required to cope with these changes. Therefore, the components of the big data computing architecture must be carefully designed considering the characteristics, cost, speed, and scalability of the system [68]. Three following types of cloud computing service models can well solve these above problems.

(1) Software as a Service (SaaS): Users can obtain data processing frameworks for different requirements with the cloud computing services. Users do not need to maintain softwares, but only need to deliver the big data to be processed to the specific SaaS cloud services, and then pay after completing the task on demand [69].

(2) Platform as a Service (PaaS): The PaaS system provides a scalable, distributed, and fault-tolerant cloud service programming platform for big data processing.

(3) Infrastructure as a Service (IaaS): IaaS cloud computing service providers provide users with configurable computing resources (processors, storages, networks and applications and I/O devices, etc.), namely virtualized resource pools. It can deal with the variability, scalability, reliability and efficiency in big data operations.

In recent years, cloud computing service model based on big data technologies has attracted a lot of research attentions. With the increasing scale of internet applications, cloud computing becomes more and more important for big data proccessing. [70] proposed a new SaaS (software as a service) design and developed a semantic model to guide the data collection process. To meet the new requirements of tenant data replication protection in SaaS, Li et al. [71] proposed a new tenant replication protection mechanism MT-DIPS based on tuple sampling. [72] discussed a context-aware safe strategy model, which can be customized according to the specific requirements of PaaS-based applications.

4.2 Big Data and Cloud Computing

Big data cannot be separated from the cloud computing. At present, the well-known cloud service providers include Amazon Web Services, Microsoft Azure, Aliyun, etc. These cloud service providers integrate data computing, algorithm development, data service and other technologies according to various data development needs, which realize a complete set of big data integration development environment on the cloud [73-74]. In terms of data processing, batch processing and stream processing are combined to process real-time data streams and historical data collaboratively. In terms of algorithm development, machine learning platform supports the implementation of regression, classification, clustering and other algorithms, and it supports popular deep learning frameworks such as TensorFlow, MXNet, Caffe, PyTorch. In addition, there are data maps in the data cloud service ecosystem, which can search data information and make data development and maintenance easier. The financial data security system on the cloud that passed the professional level test provides comprehensive data management and security solutions (e.g., data identification, sensitive data discovery, access monitoring, risk detection and anomaly detection) [75].

The fusion of big data and cloud computing technologies has been paid increasing attention by researchers. In the field of data collection and storage, Sookhak et al. [76] proposed a new data structure based on the algebraic attributes of outsourced files for cloud computing. Yang et al. [77] proposed a cloud data center energy-saving storage strategy based on a novel hypergraph overlay model. For data processing, a new computing framework called Firework is proposed in [78]. And Wang et al. [79] proposed a machine learning framework for cloud computing auxiliary resource allocation. Ezenwoke et al. [80] proposed a visual visualization framework for cloud services.

5 Big Data Application Scenarios

Big data technologies has appeared in every aspect of people’s lives, and they have been applied in various industries (e.g., finance, internet, catering, medical treatment, energy, sports and entertainment) [81-86]. In the field of Internet of things, data collection technology of wireless sensor network and data processing algorithm of big data can realize practical applications such as Internet of vehicles, novel computer architectures, Indoor localization and road anomaly detection [87-96]. Figure 5 shows the application of big data in various fields.

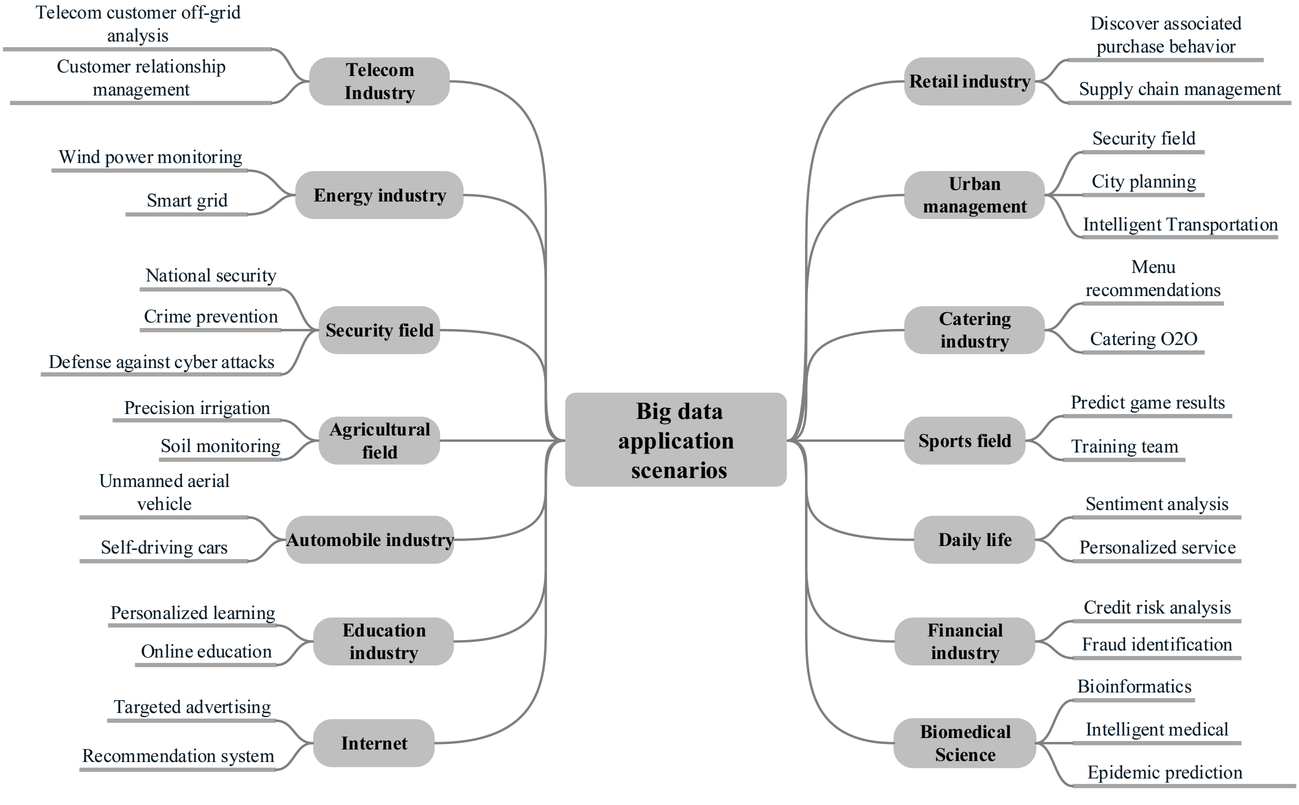

Figure 5. Big data application scenarios

5.1 Recommendation Systems

In recent years, with the exponential growth trend of all kinds of data, data consumers have to face the problem of excessive information, which makes it more difficult to make correct decisions. This phenomenon is called information overload. Recommendation system can use big data processing technologies to extract potentially valuable information from massive overload information. Its main idea is to build a model by analyzing user's historical behavior and preference information, then automatically recommend items or products of interest to users, and finally obtain personalized lists for different users [97-98].

With the exponential growth of data volume, recommendation system can intelligently analyze information and provide targeted data services for users. So this field has attracted extensive attention of researchers. [99] studied the problem of exploring the implicit hierarchy of recommendation system in the case of unclear recommendation system. And the contextual operating tensor model proposed in [100] has designed a new recommended method. In the application, Chen et al. [101] propose a time-aware smart object recommendation model by jointly in social internet of things. Due to the fact that vehicle trajectory is easily affected by the environment and user behavior, trajectory prediction is relatively low. [102] proposed a prediction model integrating environmental perception and behavioral preference. Customer information analysis and recommendation are crucial to the development of enterprises in market competition. [103] proposed an efficient query framework to find and recommend potential customers for target products.

5.2 Smart Grid

In the network of smart grid, the infrastructure provides massive information and detailed data required for automatic decision support through wireless sensor network and other technologies [104-114]. All of these data need to be processed and stored in real time for using historical or real-time data to create decisions based on certain situations [115].

Machine learning technologies can be used in smart grid to predict power consumption, pricing, power generation estimation, fault detection, adaptive control and so on [116]. The existing price forecasting, power load forecasting and other methods may be difficult to process the huge price data in the power grid. To solve these problems, a new electricity price prediction model was proposed in [117]. At the same time, in order to study the problem of data upload in the communication system between decentralized devices, Li et al. [118] proposed an optimal algorithm of polynomial running time. In addition, in other applications of smart grid, [119-121] proposed some optimization methods.

5.3 Emotional Analysis

Due to the rise of social networks, in the era of big data a new scene of large-scale data with emotional analysis arises. Emotional analysis is a process of analyzing people's emotions, opinions and evaluations, which is used to extract valuable information from massive data [122].

In massive data, high-precision emotional classification is a major challenge in emotional analysis. Recent works have effectively explored different emotion classification techniques, from simple rule-based and dictionary-based approaches to more complicated machine learning algorithms. Methods based on machine learning mainly uses machine learning algorithm related to emotion analysis to process data, such as ANN, Random Forests, Genetic Algorithm, k-nearest neighbor Algorithm, Support Vector Machine(SVM) and other algorithms. By demonstrating a model with audio, video and text as sources of information, Poria et al. [123] proposed a new multi-mode affective analysis method to obtain emotions from video networks.

6 Conclusion

With the rapid development of modern information technologies, data has become an important basis for the development of production materials and technologies. This paper investigated the big data service architecture, cloud computing services based on big data and some current big data application scenarios. First, we investigated big data infrastructures that support the big data collection and storage technologies and tools. Then according to the different data processing modes, the technical frameworks of four typical data processing were briefly introduced. Afterwards, this paper introduced the big data applications, including machine learning technologies for deep big data analysis. In addition, we discussed big data visualization technologies. Later, we introduced the cloud computing service models, as well as the combination of cloud computing and big data technologies. Finally, this paper summarized some practical application scenarios of big data technologies.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61772454, 61802031, 61811530332, 61811540410). It was also supported by the open research fund of Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education (No. JZNY201905).

References

[1] D. Goldston, Big Data: Data Wrangling, *Nature*, Vol. 455, No. 7209, pp. 15, September, 2008.

[2] A. Oguntimilehin, E. O. Ademola, A Review of Big Data Management, Benefits and Challenges, *Journal of Emerging Trends in Computing and Information Sciences*, Vol. 5, No. 6, pp. 433-438, June, 2014.

[3] V. Snášel, J. Nowaková, F. Xhafa, L. Barolli, Geometrical and Topological Approaches to Big Data, *Future Generation Computer Systems*, Vol. 67, pp. 286-296, February, 2017.

[4] J. Liu, E. Pacitti, P. Valduriez, A Survey of Scheduling Frameworks in Big Data Systems, *International Journal of Cloud Computing*, Vol. 7, No. 2, pp. 103-128, January, 2018.

[5] Y. Chen, M. Zhou, Z. Zheng, Learning Sequence-Based Fingerprint for Magnetic Indoor Positioning System, *IEEE Access*, Vol. 7, pp. 163231-163244, November, 2019.

[6] G. Bello-Orgaz, J. J. Jung, D. Camacho, Social Big Data: Recent Achievements and New Challenges, *Information Fusion*, Vol. 28, pp. 45-59, March, 2016.

[7] Y. Lin, H. Wang, J. Li, H. Gao, Data Source Selection for Information Integration in Big Data Era, *Information Sciences*, Vol. 479, pp. 197-213, April, 2019.

[8] T. Y. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, C. M. Chen, An Authenticated Key Exchange Protocol for Multi-server Architecture in 5G Networks, *IEEE Access*, Vol. 8, pp. 28096-28108, January, 2020.

[9] P. Karunaratne, S. Karunasekera, A. Harwood, Distributed Stream Clustering Using Micro-clusters on Apache Storm, *Journal of Parallel and Distributed Computing*, Vol. 108, pp. 74-84, October, 2017.

[10] J. C. Nwokeji, F. Aqlan, A. Apoorva, A. Olagunju, Big Data ETL Implementation Approaches: A Systematic Literature Review, *International Conference on Software Engineering and Knowledge Engineering (SEKE)*, Redwood, California, USA, 2018, pp. 714-715.

[11] Apache Flume, http://flume.apache.org/.

[12] B. Shu, H. Chen, M. Sun, Dynamic Load Balancing and Channel Strategy for Apache Flume Collecting Real-Time Data Stream, *IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, Guangzhou, China, 2017, pp. 542-549.

[13] Apache Kafka, http://kafka.apache.org/.

[14] H. Jafarpour, R. Desai, D. Guy, KSQL: Streaming SQL Engine for Apache Kafka, *International Conference on Extending Database Technology (EDBT)*, Lisbon, Portugal, 2019, pp. 524-533.

[15] C. A. D. Deagustini, S. E. F. Dalibón, S. Gottifredi, M. A. Falappa, C. I. Chesnevar, G. R. Simari, Relational Databases as a Massive Information Source for Defeasible Argumentation, *Knowledge-Based Systems*, Vol. 51, pp. 93-109, October, 2013.

[16] S. Ghemawat, H. Gobioff, S. T. Leung, The Google File System, *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (SOSP)*, Bolton Landing, New York, USA, 2003, pp. 29-43.

[17] M. Wang, B. Li, Y. Zhao, G. Pu, Formalizing Google File System, *IEEE 20th Pacific Rim International Symposium on Dependable Computing (PRDC)*, Singapore, Singapore, 2014, pp. 190-191.

[18] T. Yeh, T. Chien, Building a Version Control System in the Hadoop HDFS, *Network Operations and Management Symposium (NOMS)*, Taipei, Taiwan, 2018, pp. 1-5.

[19] Apache Hadoop, http://hadoop.apache.org/.

[20] K. Bok, H. Oh, J. Lim, Y. Pae, H. Choi, B. Lee, J. Yoo, An Efficient Distributed Caching for Accessing Small Files in HDFS, *Cluster Computing*, Vol. 20, No. 4, pp. 3579-3592, December, 2017.

[21] Redis, https://redis.io/.

[22] Memcached, http://memcached.org/.

[23] MongoDB, http://mongodb.org/.

[24] CouchDB, http://couchdb.apache.org/.

[25] HBase, http://hbase.apache.org/.

[26] Cassandra, http://cassandra.apache.org/.

[27] J. Hölsch, T. Schmidt, M. Grossniklaus, On the Performance of Analytical and Pattern Matching Graph Queries in Neo4j and a Relational Database, *Joint Conference: 6th International Workshop on Querying Graph Structured Data (EDBT/ICDT)*, Venice, Italy, 2017, pp. 1-8.

[28] GraphDB, http://www.sones.com.

[29] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, R. E. Gruber, Bigtable: A Distributed Storage System for Structured Data, *ACM Transactions on Computer Systems*, Vol. 26, No. 2, pp. 1-26, June, 2008.

[30] A. Pavlo, M. Aslett, What's Really New with NewSQL?, *ACM Sigmod Record*, Vol. 45, No. 2, pp. 45-55, June, 2016.

[31] J. Chen, S. Jindel, R. Walzer, R. Sen, N. Jimsheleishvilli, M. Andrews, The MemSQL Query Optimizer: A Modern Optimizer for Real-time Analytics in a Distributed Database, *Proceedings of the VLDB Endowment*, Vol. 9, No. 13, pp. 1401-1412, September, 2016.

[32] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, D. Woodford, Spanner: Google's Globally-distributed Database, *ACM Transactions on Computer Systems*, Vol. 31, No. 3, pp. 8:1-8:22, August, 2013.

[33] J. Dean, S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, *Communications of the ACM*, Vol. 51, No. 1, pp. 107-113, January, 2008.

[34] Y. Guo, J. Rao, D. Cheng, X. Zhou, iShuffle: Improving Hadoop Performance with Shuffle-on-Write, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 28, No. 6, pp. 1649-1662, June, 2017.

[35] X. Zhao, J. Zhang, X. Qin, kNN-DP: Handling Data Skewness in kNN Joins Using MapReduce, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 29, No. 3, pp. 600-613, March, 2018.

[36] Apache storm, http://storm.apache.org/.

[37] H. Yan, D. Sun, S. Gao, Z. Zhou, Performance Analysis of Storm in a Real-World Big Data Stream Computing Environment, *International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, Edinburgh, UK, 2017, pp. 624-634.

[38] R. S. Gallardo, C. Bonacic, M. Marin, S4 Applications Simulator for Performance Evaluation, *International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, Crete, Greece, 2016, pp. 403-407.

[39] L. Aniello, R. Baldoni, L. Querzoni, Adaptive Online Scheduling in Storm, *Proceedings of the 7th ACM international conference on Distributed event-based systems (DEBS)*, Arlington, Texas, USA, 2013, pp. 207-218.

[40] C. K. Shieh, S. W. Huang, L. D. Sun, M. F. Tsai, N. Chilamkurti, A Topology-based Scaling Mechanism for Apache Storm, *International Journal of Network Management*, Vol. 27, No. 3, pp. e1933, May-June, 2017.

[41] Apache Samza, http://samza.apache.org/.

[42] Z. Zhuang, T. Feng, Y. Pan, H. Ramachandra, B. Sridharan, Effective Multi-stream Joining in Apache Samza Framework, *IEEE International Congress on Big Data (BigData Congress)*, San Francisco, CA, USA, 2016, pp. 267-274.

[43] M. A. Kleppmann, J. Kreps, Kafka, Samza and the Unix Philosophy of Distributed Data, *IEEE Data Engineering Bulletin*, Vol. 38, No. 4, pp. 4-14, December, 2015.

[44] Apache Spark, http://spark.apache.org/.

[45] O. Backhoff, E. Ntoutsi, Scalable Online-Offline Stream Clustering in Apache Spark, *IEEE 16th International Conference on Data Mining Workshops (ICDM)*, Barcelona, Spain, 2016, pp. 37-44.

[46] S. Han, W. Choi, R. Muwafiq, Y. Nah, Impact of Memory Size on Bigdata Processing based on Hadoop and Spark, *Proceedings of the International Conference on Research in Adaptive and Convergent Systems (RACS)*, Krakow, Poland, 2017, pp. 275-280.

[47] D. Cheng, X. Zhou, Y. Wang, C. Jiang, Adaptive Scheduling Parallel Jobs with Dynamic Batching in Spark Streaming, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 29, No. 12, pp. 2672-2685, December, 2018.

[48] Apache Flink, https://flink.apache.org/.

[49] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, K. Tzoumas, Apache Flink: Stream and Batch Processing in a Single Engine, *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, Vol. 38, No. 4, pp. 28-38, December, 2015.

[50] B. Akil, Y. Zhou, U. Röhm, On the Usability of Hadoop MapReduce, Apache Spark & Apache Flink for Data Science, *IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, 2017, pp. 303-310.

[51] Z. Lin, *Principle and Application of Big Data Technology*, Posts and Telecom Press, 2015.

[52] G. Malewicz, M. H. Austern, A. J. C. Bik, J. C. Dehnert, I. Horn, N. Leiser, G. Czajkowski, Pregel: A System for Large-scale Graph Processing, *Proceedings of the ACM SIGMOD International Conference on Management of data (SIGMOD)*, Indiana, USA, 2010, pp. 135-146.

[53] S. Bhatia, R. Kumar, Review of Graph Processing Frameworks, *IEEE International Conference on Data Mining Workshops (ICDM)*, Singapore, Singapore, 2018, pp. 998-1005.

[54] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility, *Future Generation Computer Systems*, Vol. 25, No. 6, pp. 599-616, June, 2009.

[55] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the Game of Go with Deep Neural Networks and Tree Search, *Nature*, Vol. 529, No. 7587, pp. 484-489, January, 2016.

[56] H. Guo, R. Tang, Y. Ye, Z. Li, X. He, DeepFM: A Factorization-machine Based Neural Network for CTR Prediction, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, Melbourne, Australia, 2017, pp. 1725-1731.

[57] G. E. Hinton, S. Osindero, Y. W. Teh, A Fast Learning Algorithm for Deep Belief Nets, *Neural Computation*, Vol. 18, No. 7, pp. 1527-1554, July, 2006.

[58] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Computation*, Vol. 1, No. 4, pp. 541-551, December, 1989.

[59] S. He, Z. Li, Y. Tang, Z. Liao, F. Li, S. J. Lim, Parameters Compressing in Deep Learning, *CMC: Computers, Materials & Continua*, Vol. 62, No. 1, pp. 321-336, 2020.

[60] J. Zhang, S. Zhong, T. Wang, H.-C. Chao, J. Wang, Blockchain-Based Systems and Applications: A Survey, *Journal of Internet Technology*, Vol. 21, No. 1, pp. 1-14, January, 2020.

[61] Y. Niu, Z. Lu, J. R. Wen, T. Xiang, S. F. Chang, Multi-Modal Multi-Scale Deep Learning for Large-Scale Image Annotation, *IEEE Transactions on Image Processing*, Vol. 28, No. 4, pp. 1720-1731, April, 2019.

[62] L. Windrim, R. Ramakrishnan, A. Melkumyan, R. J. Murphy, A Physics-Based Deep Learning Approach to Shadow Invariant Representations of Hyperspectral Images, *IEEE Transactions on Image Processing*, Vol. 27, No. 2, pp. 665-677, February, 2018.

[63] P. Liu, J. M. Guo, C. Y. Wu, D. Cai, Fusion of Deep Learning and Compressed Domain Features for Content-Based Image Retrieval, *IEEE Transactions on Image Processing*, Vol. 26, No. 12, pp. 5706-5717, December, 2017.

[64] Z. T. Liu, M. Wu, W. H. Cao, J. W. Mao, J. P. Xu, G. Z. Tan, Speech Emotion Recognition Based on Feature Selection and Extreme Learning Machine Decision Tree, *Neurocomputing*, Vol. 273, pp. 271-280, January, 2018.

[65] V. Despotovic, O. Walter, R. Haeb-Umbach, Machine Learning Techniques for Semantic Analysis of Dysarthric Speech: An Experimental Study, *Speech Communication*, Vol. 99, pp. 242-251, May, 2018.

[66] P. Mell, T. Grance, *The NIST Definition of Cloud Computing*, NIST Special Publication 800-145, September, 2011.

[68] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, A View of Cloud Computing, *Communications of the ACM*, Vol. 53, No. 4, pp. 50-58, April, 2010.

[68] M. Arostegi, A. Torre-Bastida, M. N. Bilbao, J. D. Ser, A Heuristic Approach to the Multicriteria Design of IaaS Cloud Infrastructures for Big Data Applications, *Expert Systems*, Vol. 35, No. 5, pp. 1-12, October, 2018.

[69] L. Wang, J. Tao, M. Kunze, A. C. Castellanos, D. Kramer, W. Karl, Scientific Cloud Computing: Early Definition and Experience, *10th IEEE International Conference on High Performance Computing and Communications (HPCC)*, Dalian, China, 2008, pp. 825-830.

[70] I. L. Yen, F. Bastani, Y. Huang, Y. Zhang, X. Yao, SaaS for Automated Job Performance Appraisals Using Service Technologies and Big Data Analytics, *IEEE International Conference on Web Services (ICWS)*, Honolulu, HI, USA, 2017, pp. 412-419.

[72] L. Li, Y. Zhang, Y. Ding, MT-DIPS: a New Data Duplication Integrity Protection Scheme for Multi-tenants Sharing Storage in SaaS, *International Journal of Grid and Utility Computing*, Vol. 9, No. 1, pp. 26-36, February, 2018.

[72] Y. Verginadis, I. Patiniotakis, G. Mentzas, S. Veloudis, I. Paraskakis, Data Distribution and Encryption Modelling for PaaS-enabled Cloud Security, *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, Luxembourg, 2016, pp. 497-502.

[73] H. Li, W. Li, H. Wang, J. Wang, An Optimization of Virtual Machine Selection and Placement by Using Memory Content Similarity for Server Consolidation in Cloud, *Future Generation Computer Systems*, Vol. 84, pp. 98-107, July, 2018.

[74] H. Li, W. Li, Q. Feng, S. Zhang, H. Wang, J. Wang, Leveraging Content Similarity among VMI Files to Allocate Virtual Machines in Cloud, *Future Generation Computer Systems*, Vol. 79, pp. 528-542, February, 2018.

[75] T.-Y. Wu, C.-M. Chen, K.-H. Wang, C. Meng, E. K. Wang, A Provably Secure Certificateless Public Key Encryption with Keyword Search, *Journal of the Chinese Institute of Engineers*, Vol. 42, No. 1, pp. 20-28, January, 2019.

[76] M. Sookhak, F. R. Yu, A. Y. Zomaya, Auditing Big Data Storage in Cloud Computing Using Divide and Conquer Tables, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 29, No. 5, pp. 999-1012, May, 2018.

[77] T. Yang, H. Pen, W. Li, D. Yuan, A. Y. Zomaya, An Energy-Efficient Storage Strategy for Cloud Datacenters Based on Variable K-Coverage of a Hypergraph, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 28, No. 12, pp. 3344-3355, December, 2017.

[78] Q. Zhang, Q. Zhang, W. Shi, H. Zhong, Firework: Data Processing and Sharing for Hybrid Cloud-Edge Analytics, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 29, No. 9, pp. 2004-2017, September, 2018.

[79] J. B. Wang, J. Wang, Y. Wu, J. Y. Wang, H. Zhu, M. Lin, J. Wang, A Machine Learning Framework for Resource Allocation Assisted by Cloud Computing, *IEEE Network*, Vol. 32, No. 2, pp. 144-151, March-April, 2018.

[80] A. Ezenwoke, O. Daramola, M. Adigun, Towards a Visualization Framework for Service Selection in Cloud E-Marketplaces, *IEEE 13th World Congress on Services*, Honolulu, HI, USA, 2017, pp. 122-129.

[81] N. Fikri, M. Rida, N. Abghour, K. Moussaid, A. E. Omri, BigData and Regulation in High Frequency Trading, *Proceedings of the International Conference on Cloud and Big Data Computing (ICCBDC)*, London, UK, 2017, pp. 45-49.

[82] S. Tiwari, H. M. Wee, Y. Daryanto, Big Data Analytics in Supply Chain Management between 2010 and 2016: Insights to Industries, *Computers & Industrial Engineering*, Vol. 115, pp. 319-330, January, 2018.

[83] A. Kamilaris, A. Kartakoullis, F. X. Prenafeta-Boldú, A Review on the Practice of Big Data Analysis in Agriculture, *Computers and Electronics in Agriculture*, Vol. 143, pp. 23-37, December, 2017.

[84] X. Du, The Application of Big Data Technology in Competitive Sports Research, *International Conference on Geo-Spatial Knowledge and Intelligence (GSKI)*, Chiang Mai, Thailand, 2017, pp. 466-471.

[85] N. Singh, C. B. Kaverappa, J. D. Joshi, Data Mining for Prevention of Crimes, *International Conference on Human Interface and the Management of Information (HIMI)*, Las Vegas, NV, USA, 2018, pp. 705-717.

[86] I. Kalamaras, A. Zamichos, A. Salamanis, A. Drosou, D. D. Kehagias, G. Margaritis, S. Papadopoulos, D. Tzovaras, An Interactive Visual Analytics Platform for Smart Intelligent Transportation Systems Management, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 2, pp. 487-496, February, 2018.

[87] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, H. J. Kim, An Intelligent Data Gathering Schema with Data Fusion Supported for Mobile Sink in Wireless Sensor Networks, *International Journal of Distributed Sensor Networks*, Vol. 15, No. 3, pp. 1-9, March, 2019.

[88] J. Wang, X. Gu, W. Liu, A. K. Sangaiah, H.-J. Kim, An Empower Hamilton Loop Based Data Collection Algorithm with mobile Agent for WSNs, *Human-centric Computing and Information Sciences*, Vol. 9, No. 1, Article 18, May, 2019.

[89] J. Zhang, C. Wu, D. Yang, Y. Chen, X. Meng, HSCS: A Hybrid Shared Cache Scheduling Scheme for Multi-programmed Workloads, *Frontiers of Computer Science*, Vol. 12, No. 6, pp. 1090-1104, December, 2018.

[90] Y. Chen, J. Wang, R. Xia, Q. Zhang, Z. Cao, K. Yang, The Visual Object Tracking Algorithm Research Based on Adaptive Combination Kernel, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 10, No. 12, pp. 4855-4867, December, 2019.

[91] Y. Chen, J. Wang, S. Liu, X. Chen, J. Xiong, J. Xie, K. Yang, Multiscale Fast Correlation Filtering Tracking Algorithm Based on a Feature Fusion Model, *Concurrency and Computation-Parctice and Experience*, October, 2019, DOI:

10.1002/cpe.5533.

[92] Y. Chen, M. Zhou, Z. Zheng, M. Huo, Toward Practical Crowdsourcing-Based Road Anomaly Detection with Scale-Invariant Feature, *IEEE Access*, Vol. 7, pp. 67666-67678, May, 2019.

[93] K. Gao, F. Han, P. Dong, N. Xiong, R. Du, Connected Vehicle as a Mobile Sensor for Real Time Queue Length at Signalized Intersections, *Sensors*, Vol. 19, No. 9, pp. 2059, May, 2019.

[94] J. Wang, W. Wu, Z. Liao, A. K. Sangaiah, R. S. Sherratt, An Energy-efficient Off-loading Scheme for Low Latency in Collaborative Edge Computing, *IEEE Access*, Vol. 7, pp. 149182-149190, October, 2019.

[95] W. Li, Z. Chen, X. Gao, W. Liu, J. Wang, Multimodel Framework for Indoor Localization under Mobile Edge Computing Environment, *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp. 4844-4853, June, 2019.

[96] S. M. H. Rostami, A. K. Sangaiah, J. Wang, X. Liu, Obstacle Avoidance of Mobile Robots Using Modified Artificial Potential Field Algorithm, *EURASIP Journal on Wireless Communications and Networking*, Vol. 2019, No. 1, Article 70, December, 2019.

[97] H. Wang, N. Wang, D. Y. Yeung, Collaborative Deep Learning for Recommender Systems, *International Conference on Knowledge Discovery and Data Mining (KDD)*, Sydney, Australia, 2015, pp. 1235-1244.

[98] E. Çano, M. Morisio, Hybrid Recommender Systems: A Systematic Literature Review, *Intelligent Data Analysis*, Vol. 21, No. 6, pp. 1487-1524, November, 2017.

[99] S. Wang, J. Tang, Y. Wang, H. Liu, Exploring Hierarchical Structures for Recommender Systems, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 30, No. 6, pp. 1022-1035, June, 2018.

[100] S. Wu, Q. Liu, L. Wang, T. Tan, Contextual Operation for Recommender Systems, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 8, pp. 2000-2012, August, 2016.

[101] Y. Chen, M. Zhou, Z. Zheng, D. Chen, Time-aware Smart Object Recommendation in Social Internet of Things, *IEEE Internet of Things Journal*, December, 2019, DOI: 10.1109/JIOT.2019.2960822.

[102] Z. Xia, Z. Hu, J. Luo, UPTP Vehicle Trajectory Prediction Based on User Preference Under Complexity Environment, *Wireless Personal Communications*, Vol. 97, No. 3, pp. 4651-4665, December, 2017.

[103] B. Yin, K. Gu, X. Wei, A Cost-Efficient Framework for Finding Prospective Customers Based on Reverse Skyline Queries, *Knowledge-Based Systems*, Vol. 152, pp. 117-135, July, 2018.

[104] J.-S. Pan, L. Kong, T.-W. Sung, P.-W. Tsai, V. Snášel, A Clustering Scheme for Wireless Sensor Networks Based on Genetic Algorithm and Dominating Set, *Journal of Internet Technology*, Vol. 19, No. 4, pp. 1111-1118, July, 2018.

[105] Z. Meng, J.-S. Pan, K.-K. Tseng, PaDE: An Enhanced Differential Evolution Algorithm with Novel Control Parameter Adaptation Schemes for Numerical Optimization, *Knowledge-Based Systems*, Vol. 168, pp. 80-99, March, 2019.

[106] T.-T. Nguyen, J.-S. Pan, T.-K. Dao, An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network, *IEEE Access*, Vol. 7, pp. 75985-75998, June, 2019.

[107] J.-S. Pan, C.-Y. Lee, A. Sghaier, M. Zeghid, J. Xie, Novel Systolization of Subquadratic Space Complexity Multipliers Based on Toeplitz Matrix-Vector Product Approach, *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 27, No. 7, pp. 1614-1622, July, 2019.

[108] J.-S. Pan, L. Kong, T.-W. Sung, P.-W. Tsai, V. Snasel, Alpha-Fraction First Strategy for Hierarchical Model in Wireless Sensor Networks, *Journal of Internet Technology*, Vol. 19, No. 6, pp. 1717-1726, November, 2018.

[109] J. Wang, Y. Gao, C. Zhou, R. S. Sherratt, L. Wang, Optimal Coverage Multi-Path Scheduling Scheme with Multiple Mobile Sinks for WSNs, *CMC: Computers, Materials & Continua*, Vol. 62, No. 2, pp. 695-711, January, 2020.

[110] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, S. J. Lim, An Affinity Propagation-based Self-adaptive Clustering Method for Wireless Sensor Networks, *Sensors*, Vol. 19, No. 11, pp. 2579, June, 2019.

[111] J. Wang, C. Ju, H. Kim, R. S. Sherratt, S. Lee, A Mobile Assisted Coverage Hole Patching Scheme Based on Particle Swarm Optimization for WSNs, *Cluster Computing*, Vol. 22, No. 1, pp. 1787-1795, January, 2019.

[112] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, H.-J. Kim, Energy Efficient Routing Algorithm with Mobile Sink Support for Wireless Sensor Networks, *Sensors*, Vol. 19, No. 7, pp. 1494, April, 2019.

[113] J. Wang, Y. Gao, X. Yin, F. Li, H.-J. Kim, An Enhanced PEGASIS Algorithm with Mobile Sink Support for Wireless Sensor Networks, *Wireless Communications and Mobile Computing*, Vol. 2018, pp. 1-9, December, 2018.

[114] D. Cao, B. Zheng, B. Ji, Z. Lei, C. Feng, A Robust Distance-based Relay Selection for Message Dissemination in Vehicular Network, *Wireless Networks*, pp. 1-17, October, 2018, doi: 10.1007/s11276-018-1863-4.

[115] Q. Tang, K. Yang, D. Zhou, Y. Luo, F. Yu, A Real-Time Dynamic Pricing Algorithm for Smart Grid With Unstable Energy Providers and Malicious Users, *IEEE Internet of Things Journal*, Vol. 3, No. 4, pp. 554-562, August, 2016.

[116] Z. Xia, Z. Fang, F. Zou, J. Wang, A. K. Sangaiah, Research on defensive Strategy of Real-time Price Attack Based on Multiperson Zero-determinant, *Security and Communication Networks*, Vol. 2019, pp. 1-13, July, 2019.

[117] K. Wang, C. Xu, Y. Zhang, S. Guo, A. Y. Zomaya, Robust Big Data Analytics for Electricity Price Forecasting in the Smart Grid, *IEEE Transactions on Big Data*, Vol. 5, No. 1, pp. 34-45, March, 2019.

[118] W. Li, H. Xu, H. Li, Y. Yang, P. K. Sharma, J. Wang, S. Singh, Complexity and Algorithms for Superposed Data Uploading Problem in Networks with Smart Devices, *IEEE Internet of Things Journal*, October, 2019, doi: 10.1109/JIOT.2019.2949352.

[119] S. He, W. Zeng, K. Xie, H. Yang, M. Lai, X. Su, PPNC:

Privacy Preserving Scheme for Random Linear Network Coding in Smart Grid, *KSII Transactions on Internet & Information Systems*, Vol. 11, No. 3, pp. 1510-1532, March, 2017.

[120]Q. Tang, M. Xie, K. Yang, Y. Luo, D. Zhou, Y. Song, A Decision Function Based Smart Charging and Discharging Strategy For Electric Vehicle In Smart Grid, *Mobile Networks and Applications*, Vol. 24, No. 5, pp. 1722-1731, October, 2019.

[121]W. Li, H. Liu, J. Wang, L. Xiang, Y. Yang, An Improved Linear Kernel for Complementary Maximal Strip Recovery: Simpler and Smaller, *Theoretical Computer Science*, Vol. 786, pp. 55-66, September, 2019.

[122]K. Cheng, J. Li, J. Tang, H. Liu, Unsupervised Sentiment Analysis with Signed Social Networks, *Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, San Francisco, California, USA, 2017, pp. 3429-3435.

[123]S. Poria, E. Cambria, N. Howard, G. B. Huang, A. Hussain, Fusing Audio, Visual and Textual Clues for Sentiment Analysis from Multimodal Content, *Neurocomputing*, Vol. 174, pp. 50-59, January, 2016.

Biographies

Jin Wang received the M.S. degree from Nanjing University of Posts and Telecommunications, China in 2005. He received Ph.D. degree from Kyung Hee University, Korea in 2010. Now, he is a professor at Changsha University of Science and Technology. His research interests mainly include wireless sensor network and network security.

Yaqiong Yang is currently a graduate student at the School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, China. Her research interests include big data and storage systems.

Tian Wang received his Ph.D. degree in City University of Hong Kong in 2011. Currently, he is a professor at the College of Computer Science and Technology, Huaqiao University, China. His research interests include Internet of Things and edge computing.

R. Simon Sherratt received Ph.D. degree in video signal processing in 1996 from the University of Salford. He is currently a Senior Lecturer in Consumer Electronics and a Director for Teaching and Learning. His research topic is signal processing in consumer electronic devices concentrating on equalization and DSP architectures.

Jingyu Zhang received the Ph.D. degree in Computer Science and Technology from Shanghai Jiao Tong University in 2017. He is currently an Assistant Professor at the School of Computer & Communication Engineering, Changsha University of Science and Technology, China. His research interests include blockchain, computer architecture, and mobile networks.