Data Acquisition and Storage for Smart Manufacturing

Subjects: Engineering, Manufacturing

Contributor: Athina Tsanousa

The evolution of technology and especially the Internet of Things (IoT) has led to a new kind of manufacturing known as Smart Manufacturing. Smart Manufacturing is an application of the IoT that focuses on using inexpensive, small-sized, and smart devices that are all interconnected so that they can increase productivity and improve the health of the machines. Big Data in Smart Manufacturing systems are continuously generated data in high volumes produced by said smart devices and are available in various forms, e.g., log files, signal streams, or sensor data. A Big Data analysis system should be able to use these data in real time, as well as save them for historical analysis and long-term pattern detection.

smart manufacturing data fusion industry 4.0 multimodal sensors

1. Data Acquisition Methods and Technologies

In [1], a comprehensive review of Big Data analytics throughout the product lifecycle was made. Most notably, regarding the data acquisition phase of the lifecycle, it was acknowledged that manual-based data acquisition methods are still used in various stages of the lifecycle process, thus making the acquired data from these approaches inaccurate and untimely, and as a consequence, the decisions based on them are usually ineffective. The authors [2] suggested some challenges of data acquisition that need further research to be resolved and used smart mobile devices to provide an example on how IoT technologies can be embedded into the physical world and be able to gather data throughout the whole product lifecycle. A detailed hierarchical architecture of a smart factory was described by [3], emphasizing the need for Wireless Sensor Networks (WSNs) in a smart factory for data monitoring, acquisition, and logging. ZigBee and Bluetooth were also mentioned, on top of Radio-Frequency Identification (RFID), for real-time data acquisition and were described as good choices when it comes down to the cost of the industrial automation of wireless technologies. Furthermore, the devices responsible for data acquisition should be easy to set up and connect with interfaces capable of scaling up.

One methodology employed by [4] was a monitoring tool organised in a WSN. Part of the monitoring tool is a Data Acquisition Device (DAQ) that uses split-core current transformers, closed-loop Hall effect sensors, and a camera to create an easy and not intrusive way to collect data by monitoring the status of the machines. The researchers of the paper used multiple DAQs on the shop floor and used a central gateway to collect and organise the data into packets before they were transmitted. The WSN that was used for data extraction utilised the DIGI XBee ZigBee Radio-Frequency (RF) module. As the paper described, an OPC Unified Architecture (OPC-UA) was used. The OPC-UA provided an extensible data model, which provided the data schema. A NoSQL database was used, as it proved to be more flexible than a Structured Query Language database (SQL) because of the heterogeneity and the different data that were being generated. The authors of [5] create a Cyber–Physical System (CPS) that uses a semantic sensor network. Focus was given on the way data are gathered from the physical sensors. To manage the large volume and velocity of the data, the authors proposed an architecture in which the data are collected through an OPC-UA, an industrial M2M communication protocol. There exists a considerable body of literature on data flow models, but most notably, Reference [6] suggested frameworks that allow users to model their application via visual editors. These programs enable the users to receive data from external sources such as IoT devices and smart sensors.

In [7], the authors suggested an architecture design for a smart manufacturing system. Furthermore, they provided detailed considerations of the way a smart Manufacturing Execution System (MES) should be designed. For real-time data acquisition from the shop floor, the OPC-UA technology was proposed. The authors of [8] expanded on the topic of data transmission with the introduction of WiFi direct, 4G LTE, and Z-Wave. There was also the mention of the protocols being used with these wireless technologies, which include IPv6, MQTT, SOAP, and REST, among others. The authors also mentioned a series of compatible with Supervisory Control And Data Acquisition (SCADA) communication networks such as OPC, Open Database Connectivity (ODBC), RS232, and Dynamic Data Exchange (DDE), as well as some wireless communication standards such as the Highway Addressable Remote Transducer (HART) and ISA100.11a. The most prominent of these protocols are the MQTT and REST API. The MQTT protocol is used to acquire and transmit data from large industrial environments to the cloud where the processes producing the data can be monitored and controlled. The REST Application Programming Interface (API) provides a way to securely collect data from the IoT devices, where the data are collected in formal message arrays and the receivers split those individual messages so the producing device can be identified. SCADA systems provide a fully connected system that a manufacturer can use not only to acquire the data, but also to handle, manage, and archive them long term. Some known SCADA systems are SIMATIC SCADA Systems from Siemens, AVEVA™ Plant from Schneider Electric, Proficy HMI/SCADA from General Electric, and HMI/SCADA from ABB.

A demonstration of real-time data acquisition using the MQTT protocol was described in [9]. The authors implemented a system using temperature and humidity sensors so they could generate data to work with the protocol. For the test, they generated data for 60 s and compared the ability between the Hypertext Transfer Protocol (HTTP) and the MQTT protocol to transfer from the hardware (i.e., sensors) to the server and store them in a MySQL database. To minimise the error and loss of data, each transmission had a sequential ID so the completeness of data could be checked. A conclusion was made that the use of the MQTT protocol proved to be up to six-times faster than HTTP at sending data. On the more technical side, it was reported in [10] that depending on the application and the network coverage required to send data different, protocols may need to be used. Low-Energy (LE) Bluetooth, Near-Field Communications (NFC), and RFID, among others, are technologies used for short-range communication. As a result, industrial applications that require a broader field to be deployed need solutions that can be both energy saving while maintaining a significant coverage area. Such a technology is the Low-Power Wide-Area Network (LPWAN), which includes Sigfox, LoRa, and the Narrowband IoT (NB-IoT) [11].

A Big Data pipeline for data streaming in Industry 4.0 was described in [8][12]. Moreover, data collection and data storing tools were compared and presented. Such tools are Apache Kafka, RabbitMQ, and Amazon Kinesis, which are considered for pushing a high volume of messages that are produced from data producers (i.e., sensors) and even Apache Storm to process and discard “useless data”, which were tagged as less important or out of context.

In [13], a system that is capable of data monitoring and acquisition of a Computerised Numerical Control (CNC) machine tool in intelligent manufacturing was proposed and developed. Most notably, the authors compared the different data acquisition methods from a CNC machine, not only on the different data types that can be collected, but also the technical difficulty and implementation costs. For data acquisition, the MTConnect protocol was selected, while for the database, a system that uses the ODBC method was the choice. The machine tool networking was based on an industrial Ethernet and Transmission Control Protocol/Internet Protocol (TCP/IP) technology. Working also with CNC machines, Reference [14] provided a thorough explanation of the way a CNC machine generates data and how these data are acquired, transmitted, and stored. CNC data can be split into two main sources: controller data and external sensor data. As is usual for the sensors, the collected data contain noise from external interference, and for that reason, a necessary step is to clean and preprocess the data. For machines such as a CNC system, a high amount real-time data in the controller is required. Some of the most-used technologies in the field are Ethernet, Profinet, and EtherCAT, but in order for them to work seamlessly in the system, the sensors need to be equipped with acquisition cards.

Previous research showed that data acquisition in real time is based on the configuration of the smart environment [15][16][17]. Specifically, it was described that the first part of the data acquisition is the data collection, during which raw data are gathered using various technologies depending on the application. Moreover was proposed that Ultra-High-Frequency (UHF) RFID readers can be used to track and collect data in real time from the manufacturing process. Regarding the transmission of the collected data, it was mentioned that for real-time data such as temperature, vibration, and pressure, the Internet, wireless, and 4G methods were used. As far as non-real-time data (e.g., maintenance history) transmission are concerned, tools such Sqoop are preferred. A more in-depth look at the way RFID technology is used to collect data from the shop floor was provided by [18][19], where the authors explained step by step the system architecture they created. Data flow starts from RFID tags, which are attached to the input and output of their manufacturing section. This allows for real-time monitoring of the manufacturing process and can update individual data from each part. The authors concluded that such an architecture (RFID-based IoT solution) with the ability to closely monitor the manufacturing sections leads to improvements in the traceability, quality, and tool wear prediction.

Some authors [20][21] have also suggested Industrial Internet of Things (IIoT) architectures, where the data collection method was thoroughly described. Specifically, all kinds of manufacturing data (e.g., equipment status data, product data, or measurements) can be gathered using wired or wireless methods. The wireless methods include, mostly, as previously mentioned, RFID readers that obtain the raw data. Hive and HBase have been introduced as a distributed data storage system. One way that is considered for data transmission is the Flume interface, which forwards the data to a selected storage system.

2. Data Storage Software Solutions

Concerning data storage, a distributed approach is usually the choice where a Distributed Database System (DDBS) is used to store structured data and the Hadoop Distributed File System (HDFS) or NoSQL databases for unstructured data. Other alternatives are the C Open Source Managed Operating System (COSMOS) and Haystack. Especially for distributed file systems, the Google File System (GFS) was one of the first systems developed to handle data-intensive applications.

MongoDB, a NoSQL database, is one of the most popular databases at the moment, and the authors of [4] used it to store the sensor data that were gathered by the DAQs. As was mentioned by [7], relational databases such as MySQL are not an option due to the complicated nature of the manufacturing data. Subsequently, the authors considered a distributed database as an appropriate option because of its high performance, efficiency, and scalability.

Regarding data storage solutions being proposed, the authors in [12] listed, described, and compared commonly used technologies such as Hadoop Hive and MongoDB. A distinction was made between the data models that were used: firstly, the file system data model, for data stored in a schema-less manner and read in a structured manner with a processing time based on the processing needs of the application; secondly, document-based data model; lastly, a column-based schema. A reference was made regarding the recent data storage technologies and their capabilities to process data for critical real-time applications.

Prior research [22] suggested that shop floor data on a manufacturing site can be gathered using what is called a SCADA system. SCADA provides a singular interface, where all the gathered data collected from different smart devices can be transmitted. Among SCADA, Reference [23] mentioned the Protocol Data Unit (PDU) as an alternative data acquisition tool. It was also suggested that the combination of IoT and cloud services gives the ability for different equipment to be connected and collect huge amounts of data. To organise data in a methodical and effectual way, Database Management Systems (DBMSs) have been created. According to [16], these tools can be split into two categories, relational DBMS and non-relational DBMS. The first category includes SQL databases, meaning databases that usually store data in tables of records. Commercial solutions for SQL databases are Microsoft SQL, PostgreSQL, Oracle, and MySQL. Regarding NoSQL databases, it is possible to use various types of data such as text, binary, and records. One of the benefits of NoSQL over SQL is that it is scalable and can support huge volumes of data, making it perfect for managing data coming from sensors and smart devices. In [24][25], a straightforward software solution explaining the pros and cons of each one was provided. The most commonly used solution is Apache Hadoop, but the authors [24] also proposed other options such as: Redis, SimpleDB, CouchDB, and MongoDB, just to name a few. More research [26] described the criteria of the data model and used them to identify different models where each of the previously mentioned software solutions apply the best.

Authors such as [27] provided a more technical side to the way data being acquired and stored. Even though it was mentioned that the data were collected manually, there was an in-depth review of the storage methods. As has been previously reported by the literature [28], the most adequate NoSQL databases for real-time data storage are HBase and Cassandra. However, it was demonstrated by [29] that Online Transaction Processing (OLTP)-oriented NoSQL databases can lack the support for fast sequential access over a significant amount of data, which sometimes can prove to be a hurdle when it comes to data analytics. Hadoop BDW is proposed as the opposing solution that can handle fast sequential access.

A data storage framework was presented in [30] that can deal with various types of data collected from different devices, for instance RFID readers, monitors, or thermometers. Due to the heterogeneity and volume of data, there is no perfect method for efficiently storing and accessing them. The proposed architecture by the authors included several modules. In more detail, HDFS was used for unstructured file storage, while a database module using NoSQL and relational databases was used to manage the structured data. The authors also investigated a data storage framework capable of being a feasible solution to challenges such as a large volume of data, different data types, rapid generation of data, and the complicated requirements of data management. In detail, for structured data, a database management model was created that combined and extended multiple databases. For unstructured data, a common solution was followed. The authors explained in great detail how the framework extended HDFS for multitenant data isolation. Concluding, it was mentioned that for remote and cross-platform data access, a RESTful-service-generating mechanism was integrated, to provide a platform-independent HTTP interface. Furthermore, the authors of [31] reported that in large-scale manufacturing systems, tens of thousands of data streams flow into the storage at various rates. For that reason, there is a need for improving the bottlenecks that cause the data to arrive irregularly. The authors mentioned Blueflood as a solution that attempts to achieve high scalability. Blueflood is a combination of Cassandra, which handles data storage, and Elasticsearch, which handles indexing. A similar alternative to Blueflood is OpenTSDB, which uses HBase for storage instead of Cassandra. A summary of the software tools used for data management can be found in **Table 1**.

The authors of [32] separated the requirement and solution components that are assigned to the data storage processes. In [33], it was suggested that due to comprehensive process transparency, structured, semi-structured, and unstructured data should be stored and made available for application-specific processing. Furthermore, in [32], operational data storage and a long data storage system were proposed. The first one requires an edge device unit, which must be able to store and manage real-time data. For operational data storage, the edge devices need to store data efficiently and reliably. For that reason, SQL was recommended. The preferred Relational Database Management Systems (RDBMSs) are MySQL and PostgreSQL. Concerning long-term data storage, a Big Database system is required. Consequently, NoSQL databases provide the best storage solutions as they are able to efficiently store large volumes of unstructured datasets, compared to relational databases [24][34].

A focus on data storage issues and recommendations for a new solution to organise and manage data was given by [35]. In particular, the cloud storage system they presented uses a Document-Oriented Storage System (DO-SS) for the storage of all the information derived from the monitoring systems. The integration between the data collection and storage subsystem occurs with the help of a software module (parser), so that the data can be converted before being stored in MongoDB. The authors also implemented an Object-Oriented Storage System (OO-SS), a widely used object storage system. The main benefit over other solutions, is that the data are protected by being stored as multiple copies, so in case a node fails, there is another one active where the data are stored. This design makes the OO-SS ideal when there is a need for performance and scalability.

In [36], a hybrid framework was conceptualised for an industrial platform that ensures efficient and accurate communication, concerning data transfer among software applications and devices. The framework was characterised as hybrid, because it contains two different technologies for data storage and exploits the best features from each of them. In the proposed framework, structured data are stored in relational database systems, while sensorial data, which most of the time tend to be unstructured, are stored in NoSQL databases. The real-time sensor data are published to an MQTT broker that is suitable to be used to connect with remote locations, and the Raw Data Handler subscribes to MQTT and acquires the generated data. Later, the data are carried to the sensorial repository where they are indexed and can easily be filtered and accessed by timestamps. In general, the authors proposed a hybrid framework that is capable of shop floor data collection and application in industrial environments.

A scalable data storage framework for smart manufacturing was introduced by [37]. In more detail, it is a Software-Defined Hybrid Cloud (SDHC) for saving the equipment-generated data. The main challenge the authors faced was the different data types and formats. With the use of software-defined technology, control data and manufacturing data were separated. The manufacturing data derived from the sensors and controllers can be saved in a key-value database only if the data save request is allowed. Lastly, two improvements were proposed, the first one to deal with the data saving efficiency, which will improve the response time, and the second one with the data-save permission, which will improve the system’s robustness. A software architecture was designed by [38]. The framework is highly scalable, so a fleet of IoT boards and sensors can be easily configurable. For data collection, an Arancino board was used that was provided with an AI module that can manage on-board fault prediction. InfluxDB was the database that was selected as it is non-relational and is suitable for industrial scenarios where sensors send data at different rates.

Table 1. Applications of software tools for data management.

Software Tool	Application	Reference
Apache Hadoop	Hadoop is a framework that allows for the distributed processing of large datasets across clusters of computers using simple programming models. It has been used for different kinds of applications such as frameworks that can optimise and organise the way bit data can be searched and accessed. There are also applications regarding storing data derived from sensors that monitor the environmental air pollution. Lastly, tuning systems have been designed to improve the performance of Hadoop and MapReduce.	[39][40][41]
Apache Storm	One of the most capable software solutions for Big Data is Apache Storm. Several applications exist that employ it. Some of them use it as a data streaming and real-time processing platform, while others create frameworks for dynamically scaling for the analysis of streaming data. Finally, there are multisensor data fusion frameworks that employ Apache Storm due to its high reliability and good processing mode.	[42][43][44]

Software Tool	Application	Reference
Apache Flume	Flume is a distributed, reliable, and available system for efficiently collecting, aggregating, and moving large amounts of event data. It has been used for various kinds of applications such as healthcare and manufacturing. Frameworks have been designed so that the computational scalability of sensor network data can be achieved.	[45][46][47]
Apache Spark	The aim of Spark is to make data analytics programs run faster by offering a general execution model that optimises arbitrary operator graphs and supports in-memory computing. Most applications use it for sensor analytics. It has been deployed on both industrial and non-industrial applications and can be integrated into pre-existing frameworks.	[48][49][50]
Apache Kafka	Kafka is well suited for the situations where users need to process real-time data and analyse them. There are papers that focused on learning how to reliably transfer data and studied its application in collaboration with other software solutions.	[51][52][53]

3. Communication Protocols

The smart manufacturing sector benefits from data fusion systems. Next, the required underlying communication technologies enabling the fusion systems are presented. These are related to the IoT and the corresponding protocols. Specifically, networking concepts that are based on software are incorporated into the lower communication layers. Furthermore, adaptivity offers advanced performance since the nature of modern networking systems is dynamic.

Software-Defined Networking (SDN) is the main conceptual networking model [54] under modern IoT fusion environments. It brings the programmable networking logic into the lower architectural layers. This process allows better control and management of networking data flows in a transparent way from the higher-layer networking applications. It is a centralised architecture that defines a stable ground to be used for building networking applications. An open implementation of the SDN networking concept is the OpenFlow protocol [55], which is widely adopted. A networking foundation is currently maintaining the specification. The whole concept relies on central computing logic, represented by the SDN controller, controlling data flows between core networking components such as switches (i.e., the Southbound API). Fusion techniques over IoT environments are facilitated with the exploitation of the SDN concept.

Achieving high-speed transmissions in IoT environments requires efficient and dynamic channel assignment. Conventional fixed assignment techniques are not adequate for modern environments, in which, due to their dynamic nature, requirements must constantly adapt to the runtime conditions. The IoT over SDN, when combined with deep learning techniques, improves transmission quality. Therefore, a traffic load prediction algorithm that is based on deep learning [56] has been proposed, forecasting network traffic and congestion. Next, a deep-learning-based algorithm that assigns channels has been introduced. Its main role relates to link channel allocation using intelligence in the SDN-IoT network environment.

Since communication between smart devices in the IoT manufacturing sector can be peculiar, event-based data fusion for communication is needed [57]. This is a message exchange system between participating devices that initiates when events occur. Fusion is required since different devices generate heterogeneous notifications, along with data source trust issues that may arise. The contribution of this work consisted of an event-based protocol covering the communication issues of resource-limited sensors and heterogeneous data sources, and it considered the trust degree of the fused data.

There is a vast spectrum of IoT applications that require security and privacy for realistic deployment in the modern era. Trust and data integrity are prerequisites in the IoT ecosystem, otherwise applications will lose high demand and also their potential. In the current case of cellular and sensor networks, special security challenges emerge and correlate with authentication issues, privacy, management, and information storage [58]. Programmable Logic Controllers (PLCs) are an integral part of the industrial control systems [59]. Communication issues between PLCs and the engineering stations or field devices concerning security must be confronted. Modern database systems use communication systems to deploy as cloud-based solutions [60]. Different DBs support various security technologies, though most non-relational solutions overlook modern Big Data applications.

Communication requires a credible authentication model, which guarantees data integrity and secrecy. For that purpose, an IoT node-roaming authentication protocol was introduced [61]. A heterogeneous fusion mechanism comprises the protocol's functionality. Every roaming device communicates with a server, which provides authentication functionality. This process renders attacking attempts from external malicious nodes difficult.

In a smart manufacturing environment, multiple protocols are required for transmitting data efficiently. SDN forms the basis for a heterogeneous network architecture [62] for forwarding multisource manufacturing data and, at the same time, utilising network resources optimally. The core algorithm of the proposed protocol is based on cross-network fusion and scheduling. It was shown that the efficiency was improved for the fusion processes, especially for intelligent manufacturing equipment.

References

1. Ren, S.; Zhang, Y.; Liu, Y.; Sakao, T.; Huisingh, D.; Almeida, C.M. A comprehensive review of Big Data analytics throughout product lifecycle to support sustainable smart manufacturing: A framework, challenges and future research directions. J. Clean. Prod. 2019, 210, 1343–1365.
2. Zheng, P.; Lin, T.J.; Chen, C.H.; Xu, X. A systematic design approach for service innovation of smart product-service systems. J. Clean. Prod. 2018, 201, 657–667.
3. Chen, B.; Wan, J.; Shu, L.; Li, P.; Mukherjee, M.; Yin, B. Smart factory of industry 4.0: Key technologies, application case, and challenges. IEEE Access 2017, 6, 6505–6519.
4. Mourtzis, D.; Vlachou, E.; Milas, N. Industrial Big Data as a result of IoT adoption in manufacturing. Procedia CIRP 2016, 55, 290–295.
5. Obitko, M.; Jirkovskỳ, V. Big data semantics in industry 4.0. In Proceedings of the International Conference on Industrial Applications of Holonic and Multi-Agent Systems, Valencia, Spain, 2–3

September 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 217–229.

6. Gokalp, M.O.; Kayabay, K.; Akyol, M.A.; Eren, P.E.; Koçyiğit, A. Big data for industry 4.0: A conceptual framework. In Proceedings of the 2016 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 15–17 December 2016; pp. 431–434.
7. Jeon, B.W.; Um, J.; Yoon, S.C.; Suk-Hwan, S. An architecture design for smart manufacturing execution system. Comput.-Aided Des. Appl. 2017, 14, 472–485.
8. Saqlain, M.; Piao, M.; Shim, Y.; Lee, J.Y. Framework of an IoT-based industrial data management for smart manufacturing. J. Sens. Actuator Netw. 2019, 8, 25.
9. Atmoko, R.; Riantini, R.; Hasin, M. IoT real time data acquisition using MQTT protocol. J. Phys. Conf. Ser. 2017, 853, 012003.
10. Xu, J.; Yao, J.; Wang, L.; Ming, Z.; Wu, K.; Chen, L. Narrowband internet of things: Evolutions, technologies, and open issues. IEEE Internet Things J. 2017, 5, 1449–1462.
11. Mekki, K.; Bajic, E.; Chaxel, F.; Meyer, F. A comparative study of LPWAN technologies for large-scale IoT deployment. ICT Express 2019, 5, 1–7.
12. Sahal, R.; Breslin, J.G.; Ali, M.I. Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case. J. Manuf. Syst. 2020, 54, 138–151.
13. Guo, Y.; Sun, Y.; Wu, K. Research and development of monitoring system and data monitoring system and data acquisition of CNC machine tool in intelligent manufacturing. Int. J. Adv. Robot. Syst. 2020, 17, 1–12.
14. Xiao, Y.; Liu, Q. Application of Big Data processing method in intelligent manufacturing. In Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2019; pp. 1895–1900.
15. Zhang, Y.; Ren, S.; Liu, Y.; Si, S. A Big Data analytics architecture for cleaner manufacturing and maintenance processes of complex products. J. Clean. Prod. 2017, 142, 626–641.
16. Dai, H.N.; Wang, H.; Xu, G.; Wan, J.; Imran, M. Big data analytics for manufacturing internet of things: Opportunities, challenges and enabling technologies. Enterp. Inf. Syst. 2020, 14, 1279–1303.
17. Majeed, A.; Zhang, Y.; Ren, S.; Lv, J.; Peng, T.; Waqar, S.; Yin, E. A Big Data-driven framework for sustainable and smart additive manufacturing. Robot.-Comput.-Integr. Manuf. 2021, 67, 102026.
18. Srinivasan, M.; Prince, E.; Padmanabhan, R. IoT architecture for advanced manufacturing technologies. Mater. Today Proc. 2020, 22, 2359–2365.

19. Meng, Z.; Wu, Z.; Gray, J. RFID-based object-centric data management framework for smart manufacturing applications. IEEE Internet Things J. 2018, 6, 2706–2716.
20. Wan, J.; Tang, S.; Shu, Z.; Li, D.; Wang, S.; Imran, M.; Vasilakos, A.V. Software-defined industrial internet of things in the context of industry 4.0. IEEE Sens. J. 2016, 16, 7373–7380.
21. Sanghavi, D.; Parikh, S.; Raj, S.A. Industry 4.0: Tools and implementation. Manag. Prod. Eng. Rev. 2019, 10, 3–13.
22. Frank, A.G.; Dalenogare, L.S.; Ayala, N.F. Industry 4.0 technologies: Implementation patterns in manufacturing companies. Int. J. Prod. Econ. 2019, 210, 15–26.
23. Zhang, X.; Ming, X.; Liu, Z.; Qu, Y.; Yin, D. An overall framework and subsystems for smart manufacturing integrated system (SMIS) from multi-layers based on multi-perspectives. Int. J. Adv. Manuf. Technol. 2019, 103, 703–722.
24. Gölzer, P.; Cato, P.; Amberg, M. Data Processing Requirements of Industry 4.0-Use Cases for Big Data Applications. In Proceedings of the Twenty-Third European Conference on Information Systems (ECIS), Münster, Germany, 26–29 May 2015; Paper 61.
25. Lade, P.; Ghosh, R.; Srinivasan, S. Manufacturing analytics and industrial internet of things. IEEE Intell. Syst. 2017, 32, 74–79.
26. Pokorny, J. NoSQL databases: A step to database scalability in web environment. Int. J. Web Inf. Syst. 2013, 9, 69–82.
27. Santos, M.Y.; e Sá, J.O.; Andrade, C.; Lima, F.V.; Costa, E.; Costa, C.; Martinho, B.; Galvão, J. A Big Data system supporting bosch braga industry 4.0 strategy. Int. J. Inf. Manag. 2017, 37, 750–760.
28. Costa, C.; Santos, M.Y. Reinventing the energy bill in smart cities with NoSQL technologies. In Transactions on Engineering Technologies; Springer: Berlin/Heidelberg, Germany, 2016; pp. 383–396.
29. Costa, C.; Santos, M.Y. The SusCity Big Data warehousing approach for smart cities. In Proceedings of the 21st International Database Engineering & Applications Symposium, Bristol, UK, 12–14 July 2017; pp. 264–273.
30. Jiang, L.; Da Xu, L.; Cai, H.; Jiang, Z.; Bu, F.; Xu, B. An IoT-oriented data storage framework in cloud computing platform. IEEE Trans. Ind. Inform. 2014, 10, 1443–1451.
31. Yen, I.L.; Zhang, S.; Bastani, F.; Zhang, Y. A framework for IoT-based monitoring and diagnosis of manufacturing systems. In Proceedings of the 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE), San Francisco, CA, USA, 6–9 April 2017; pp. 1–8.
32. Vater, J.; Harscheidt, L.; Knoll, A. A reference architecture based on edge and cloud computing for smart manufacturing. In Proceedings of the 2019 28th International Conference on Computer

Communication and Networks (ICCCN), Valencia, Spain, 29 July–1 August 2019; pp. 1–7.

33. Krumeich, J.; Werth, D.; Loos, P. Prescriptive control of business processes. Bus. Inf. Syst. Eng. 2016, 58, 261–280.
34. Raghav, R.; Amudhavel, J.; Dhavachelvan, P. A survey of NoSQL database for analysing large volume of data in big data platform. Int. J. Eng. Technol. (UAE) 2018, 7, 181–186.
35. Fazio, M.; Celesti, A.; Puliafito, A.; Villari, M. Big data storage in the cloud for smart environment monitoring. Procedia Comput. Sci. 2015, 52, 500–506.
36. Grevenitis, K.; Psarommatis, F.; Reina, A.; Xu, W.; Tourkogiorgis, I.; Milenkovic, J.; Cassina, J.; Kiritsis, D. A hybrid framework for industrial data storage and exploitation. Procedia CIRP 2019, 81, 892–897.
37. Wang, H.Y.; Tsung, C.K. Scalable Data-Storage Framework for Smart Manufacturing. In Proceedings of the International Conference on Frontier Computing, Ischia, Italy, 8–10 May 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 307–313.
38. De Vita, F.; Bruneo, D.; Das, S.K. On the use of a full stack hardware/software infrastructure for sensor data fusion and fault prediction in industry 4.0. Pattern Recognit. Lett. 2020, 138, 30–37.
39. Ghaemi, Z.; Farnaghi, M.; Alimohammadi, A. Hadoop-based distributed system for online prediction of air pollution based on support vector machine. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 2015, 1, 215–219.
40. Abdelouarit, K.A.; Sbihi, B.; Aknin, N. Towards an approach based on hadoop to improve and organize online search results in Big Data environment. In ICCMIT 2016, Communication, Management and Information Technology; CRC Press: Cosenza, Italy, 2016; pp. 557–564.
41. Ding, X.; Liu, Y.; Qian, D. Jellyfish: Online performance tuning with adaptive configuration and elastic container in hadoop yarn. In Proceedings of the 2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS), Melbourne, Australia, 14–17 December 2015; pp. 831–836.
42. Iqbal, M.H.; Soomro, T.R. Big data analysis: Apache storm perspective. Int. J. Comput. Trends Technol. 2015, 19, 9–14.
43. Van Der Veen, J.S.; Van Der Waaij, B.; Lazovik, E.; Wijbrandi, W.; Meijer, R.J. Dynamically scaling apache storm for the analysis of streaming data. In Proceedings of the 2015 IEEE First International Conference on Big Data Computing Service and Applications, Washington, DC, USA, 30 March–2 April 2015; pp. 154–161.
44. Yan, L.; Shuai, Z.; Bo, C. Multisensor data fusion system based on Apache Storm. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 1094–1098.

45. Rashid, M.; Singh, H.; Goyal, V.; Parah, S.A.; Wani, A.R. Big data based hybrid machine learning model for improving performance of medical Internet of Things data in healthcare systems. In Healthcare Paradigms in the Internet of Things Ecosystem; Elsevier: Amsterdam, The Netherlands, 2021; pp. 47–62.
46. Manogaran, G.; Lopez, D. Health data analytics using scalable logistic regression with stochastic gradient descent. Int. J. Adv. Intell. Paradig. 2018, 10, 118–132.
47. Makeshwar, P.; Kalra, A.; Rajput, N.; Singh, K. Computational scalability with Apache Flume and Mahout for large scale round the clock analysis of sensor network data. In Proceedings of the 2015 National Conference on Recent Advances in Electronics & Computer Engineering (RAECE), Roorkee, India, 13–15 February 2015; pp. 306–311.
48. Shi, W.; Zhu, Y.; Huang, T.; Sheng, G.; Lian, Y.; Wang, G.; Chen, Y. An integrated data preprocessing framework based on apache spark for fault diagnosis of power grid equipment. J. Signal Process. Syst. 2017, 86, 221–236.
49. Shyam, R.; HB, B.G.; Kumar, S.; Poornachandran, P.; Soman, K. Apache spark a Big Data analytics platform for smart grid. Procedia Technol. 2015, 21, 171–178.
50. Jayaratne, M.; Alahakoon, D.; De Silva, D.; Yu, X. Apache spark based distributed self-organizing map algorithm for sensor data analysis. In Proceedings of the IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November 2017; pp. 8343–8349.
51. Wu, H.; Shang, Z.; Wolter, K. Learning to reliably deliver streaming data with apache kafka. In Proceedings of the 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Valencia, Spain, 29 June–2 July 2020; pp. 564–571.
52. Kato, K.; Takefusa, A.; Nakada, H.; Oguchi, M. A study of a scalable distributed stream processing infrastructure using Ray and Apache Kafka. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 5351–5353.
53. Wu, H.; Shang, Z.; Peng, G.; Wolter, K. A reactive batching strategy of apache kafka for reliable stream processing in real-time. In Proceedings of the 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE), Coimbra, Portugal, 12–15 October 2020; pp. 207–217.
54. Das, T.; Sridharan, V.; Gurusamy, M. A survey on controller placement in SDN. IEEE Commun. Surv. Tutor. 2019, 22, 472–503.
55. Alsaeedi, M.; Mohamad, M.M.; Al-Roubaiey, A.A. Toward adaptive and scalable OpenFlow-SDN flow control: A survey. IEEE Access 2019, 7, 107346–107379.
56. Tang, F.; Fadlullah, Z.M.; Mao, B.; Kato, N. An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach. IEEE Internet Things J. 2018, 5, 5141–5154.

57. Esposito, C.; Castiglione, A.; Palmieri, F.; Ficco, M.; Dobre, C.; Iordache, G.V.; Pop, F. Event-based sensor data exchange and fusion in the Internet of Things environments. J. Parallel Distrib. Comput. 2018, 118, 328–343.

58. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A survey on IoT security: Application areas, security threats, and solution architectures. IEEE Access 2019, 7, 82721–82743.

59. Ghaleb, A.; Zhioua, S.; Almulhem, A. On PLC network security. Int. J. Crit. Infrastruct. Prot. 2018, 22, 62–69.

60. Samaraweera, G.D.; Chang, J.M. Security and privacy implications on database systems in Big Data era: A survey. IEEE Trans. Knowl. Data Eng. 2019, 33, 239–258.

61. Wan, Z.; Xu, Z.; Liu, S.; Ni, W.; Ye, S. An internet of things roaming authentication protocol based on heterogeneous fusion mechanism. IEEE Access 2020, 8, 17663–17672.

62. Wan, J.; Yang, J.; Wang, S.; Li, D.; Li, P.; Xia, M. Cross-network fusion and scheduling for heterogeneous networks in smart factory. IEEE Trans. Ind. Inform. 2019, 16, 6059–6068.

Retrieved from https://encyclopedia.pub/entry/history/show/48852