Hindawi
Mobile Information Systems
Volume 2021, Article ID 8557690, 13 pages
https://doi.org/10.1155/2021/8557690

Research Article

Design and Implementation of Data Processing Software for Internet of Things Based on Virtual Reality

Yuwang Liu,[1] **Xiaoming Liu**,[2] **and Shaotong Xu**[1]

[1]*Information Technology and Equipment Center, Guangzhou College of Technology and Business, Guangzhou, Guangdong 510800, China*
[2]*Office of Informatization Management, Sun Yat-Sen University, Guangzhou, Guangdong 510275, China*

Correspondence should be addressed to Yuwang Liu; jon@gzgs.edu.cn

Received 9 August 2021; Revised 2 September 2021; Accepted 13 September 2021; Published 20 October 2021

Academic Editor: Sang-Bing Tsai

With the rapid development of the times, the application of the Internet of Things also develops rapidly, resulting in a linear increase in the data generated by the application. Nowadays, people's needs for the forms and types of data are becoming more and more diversified, and the demand for the analysis and processing of Internet of Things data is only increasing. In order to meet the demand for data analysis of the Internet of Things, this article mainly introduces the design and implementation of the data analysis and processing software of the Internet of Things based on virtual reality. Realistic data analysis algorithms are applied to the data analysis function modules of the Internet of Things data analysis and processing software, and then the design of the data collection, data capture, and data analysis and processing functions of the Internet of Things data analysis and processing system is improved. Finally. The realization of the data analysis and processing software of the Internet of Things is carried out through different environments for and various aspects of software testing. The test results show that the realization rate of the software function is as high as 90%, and the data loss rate is reduced to 5%.

1. Introduction

1.1. Research Background and Significance. With the rapid development of the Internet of Things, large-scale construction of application systems and Internet of Things equipment in various fields has encountered operational difficulties, high costs, low efficiency, information islands, and scattered applications in the later stages of construction. The combination of virtual reality technology and Internet of Things technology provides a broader development space for the Internet of Things [1]. The virtual reality technology perceives the network and, on this basis, develops various intelligent management and service systems according to the actual needs of the network field, which promotes the business development and development of the Internet of Things. The amount of data will increase exponentially, and the network access time and control corresponding time requirements will reach the millisecond level. Mass data analysis and processing will become inevitable, requiring related technology upgrades. The data analysis and processing system is an important part of the Internet of Things, which can provide real-time data for management and control and provide an important basis for equipment monitoring and performance analysis for operators [2]. In order to shorten the system development cycle and avoid the repeated design of the system, the design and research of the data analysis and processing software of the Internet of Things are of great significance. In the application of the Internet of Things, analog signals and digital signals are usually analyzed and processed. The current data analysis and processing systems of the Internet of Things are generally specially developed for a certain system or a certain industry [3]. Although it has strong functions, it also has shortcomings such as poor versatility, high price, waste of resources, and weak secondary development capabilities. Moreover, there are various interferences in the operating environment, leading to deviations in data analysis, leading to varying degrees of consequences. Therefore, it is of great significance to design a universal and reliable IoT data analysis and processing software.

1.2. Related Content. Some immersive VR dispersions can reduce the intensity of subjective pain caused by nociception in multimodal experiments. Sharar S. R. put forward a subjective report that immersive virtual reality (VR) dispersion therapy can effectively relieve pain in the clinic and increase "fun" in the medical environment of procedural pain [4]. The purpose of his research is to use the loop model (pleasure/stimulation) to better describe the "fun" variables related to VR distraction and analgesia. He believes that, during VR distraction, there are many factors related to subjective analgesia, including lower anxiety, more fun, activeness in the VR environment, and positive emotional valence [5, 6]. Subjects who reported less anxiety, more fun, more VR presence, and more positive emotional valence in VR distraction were more likely to report a reduction in subjective pain. His findings indicate that VR dispersive analgesia can be mediated through antianxiety, attention, or affective mechanisms. But the applicability of this is narrow. With the popularity of the Internet of Things (IoT), data has exploded. Data analysis is the basis of IoT-based applications, and clusters are an important tool for data analysis [7, 8]. In clustering, determining the number of clusters is an important issue, which can be manually specified or automatically determined. The manual method has many disadvantages. The automatic method has obvious advantages, and its key task is to design a suitable cluster update algorithm. Although many studies have been conducted, most of the studies are still invalid or cannot guarantee unique clustering results and good clustering accuracy [9]. At the same time, considering that IoT-based applications always involve both digital data and nonnumerical data and it is impractical to process all nonnumerical data in the same way, X. Yao tried to further classify nonnumerical attributes according to their nature and, respectively, explore the corresponding similarity indicators [10]. On this basis, according to the dissimilarity and density of the data objects, an algorithm to determine the initial cluster center is proposed. Then, for the mixed data, an improved clustering algorithm was designed based on the revised intercluster entropy. But this algorithm has errors. The Internet of Things (IoT) plays a vital role in shaping today's technological world [11]. Palaniswami M. introduced a set of useful visual assessment clustering trend (VAT) tools and techniques [12]; these tools and techniques have made a significant contribution to software design and emphasized how these technologies can be achieved through the large-scale Internet of Things to promote the development of the Internet of Things. He discovered from these ubiquitous heterogeneous activity resources and equipment that integrating and interpreting the processed big data is a difficult task [13]. Therefore, the cluster analysis of big data generated by the Internet of Things is essential for meaningful interpretation of such complex data. However, the accuracy of cluster analysis of the Internet of Things is still lacking in discussion.

1.3. Main Content and Innovation. The main content of this article is to design and implement software based on the analysis and processing of IoT data based on virtual reality technology, provide algorithm technology for data analysis through virtual reality technology, and then analyze and process the data of the Internet of Things. Functions such as acquisition module, data capture module, data analysis module, and data storage module are designed, and then the functions designed by the software are tested through software operation, thereby realizing IoT data analysis and processing software. The innovation of this article lies in the use of virtual reality technology to provide certain data analysis calculations for the Internet of Things data analysis software and provide unique technical support for the Internet of Things data analysis software and then realize the data analysis and processing software of the Internet of things.

2. Data Analysis Technology

2.1. Data Analysis and Calculation of Virtual Reality Technology. In the optimized configuration of the virtual machine CPU, it is mentioned that if the application is not multithreaded, in order to improve the running performance of the application, you can consider using multiple virtual machines (horizontal expansion) [14, 15]. Even for multithreaded applications, the coordination of multiple virtual machines and hardware load balancing is more conducive to the improvement of application system performance. It can also improve the efficiency of data processing. The frequency domains can be converted to each other with the help of Fourier transform and time domain [16]. The relationship is as follows:

$$X(f) = \int_{-1}^{1} x(t) \cdot e dt, \tag{1}$$

$$x(t) = \int_{-1}^{1} X(f) e df. \tag{2}$$

Among them, F is the Fourier transform, formula (2) is the inverse Fourier transform, x is the time-domain data, and f is the corresponding frequency-domain data. It can be seen from the above formulas that if you want to calculate the data accurately, you need to obtain the signal indefinitely, which is obviously impossible in practice [17, 18]. Therefore, in practice, the Fourier transform uses data signal processing to sample and quantize the analyzed signal. In order to prevent the loss of time domain data, the sampling rate must meet the sampling law [19, 20]. The calculation of the frequency spectrum of the sampling point is called the discrete Fourier transform, and formulas (1) and (2) become

$$\begin{aligned} X(k) &= \sum_{n=0}^{n} x(n).W, \\ x(n) &= \frac{1}{n} \sum_{k=0}^{n} x(k) \cdot W, \end{aligned} \tag{3}$$

where N is the sampling length, x is the sampling sequence of time-domain data, set the sequence length to N and satisfy $N = 2$, and M is a natural number, divided into two subsequences of n according to the parity of n [21, 22].

$$x_1(i) = x(2i),$$
$$x_2(i) = x(2i+1). \tag{4}$$

Then the DFT of x is

$$X(k) = \sum_{n=0}^{n} x(n) \cdot w. \tag{5}$$

Since x has an implicit periodicity [23, 24] and the period is N|2, we have

$$X_1 = \left(k + \frac{n}{2}\right) = X_1(k). \tag{6}$$

The ratio of complex number addition is

$$R = \frac{n(n-1)}{n \log_2 n}. \tag{7}$$

Figure 1 shows the comparison curve between the ratio of the number of multiplications and additions required by the direct use of DFT and FFT algorithms and the number of points N [25, 26]. The processor's operation efficiency for addition is much higher than that of multiplication [27]. Therefore, the efficiency of the algorithm is mainly reflected in the number of multiplications [28]. It can be seen from the figure that the efficiency of complex multiplication is better than that of complex addition. The efficiency improvement rate should be fast. The efficiency of the FFT algorithm is much higher than that of the direct DFT. When N is larger, the algorithm efficiency is higher [27, 29].

3. Software Functional Requirements and Software Functional Design

3.1. IoT Framework Based on Virtual Technology. The Internet of Things has three different levels. The application layer is the layer in the actual use of the Internet of Things, which represents the functions and service requirements of the Internet of Things; the network layer includes the local area network, the division of labor and coordination, and connection support structure of the Internet; the physical layer is the real thing through different or multiple underlying structures of a medium connection. Information collection, transmission, processing, and presentation are important information processes in the Internet of Things [30, 31]. They are reflected in the sensing layer, network layer, and application layer of the Internet of Things, as shown in Figure 2.

As shown in Figure 2, the sensing layer is similar to the five senses of a person, and the function is identifying the special attribute information of the object through the device sensors such as GPS and camera, so that the information of the object can be obtained at any time. The network layer transmits data information through the mobile network and some other networks and is responsible for transmitting the

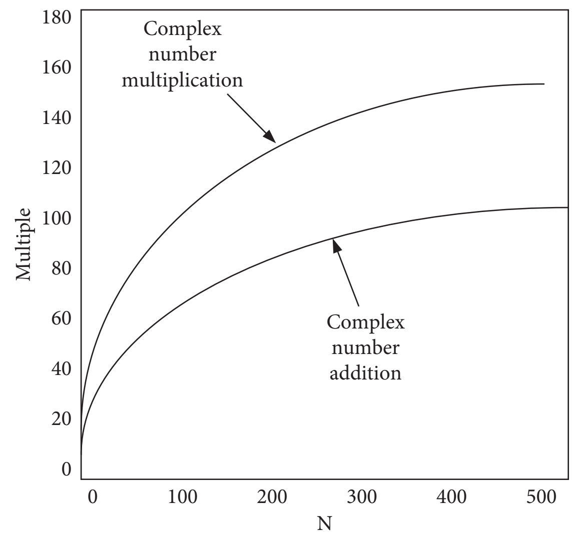

Figure 1: The relationship between the ratio of the number of complex multiplications and the number of additions and the number of points.

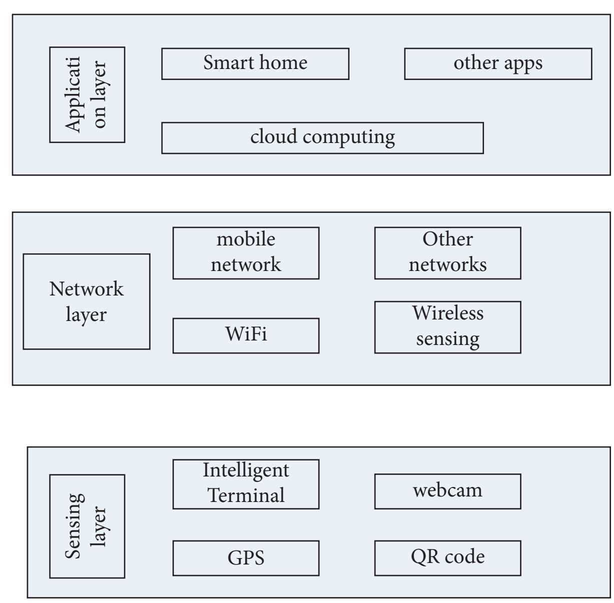

Figure 2: IoT architecture.

data information obtained by the sensor layer. The application layer is to process the data information transmitted at the network layer to make judgments and decisions [32–34]. These three levels are very important and interrelated, and none of them are indispensable.

3.2. Software Performance Requirements

3.2.1. Nonfunctional Requirements. According to the above-mentioned issues related to the Internet of Things data, the software nonfunctions meet at least the following requirements.

As shown in Table 1, the nonfunctional requirements of software include integration, integrity, consistency, access

Table 1: Software nonfunctional requirements.

Nonfunctional requirements	The reason
Integration	Internet of Things data has logic and structure; data contact is recommended
Completeness	To meet the conditions of the application processing data
Consistency	Different data sources have direct semantic differences and then a unified data structure
Access security	Data sources belong to different applications and are confidential
Optimization	To reduce waste of resources and waiting time

security, and optimization. These software performance requirements have a critical impact on software design and development to a certain extent. It can increase the experience of the software.

3.2.2. Software Function Requirements. Internet of Things data processing and analysis software is very important for analyzing the operation of Internet of Things smart devices. It not only helps technicians process and analyze recorded data but also provides interesting applications for Internet of Things smart devices, provides users with a better experience, and provides technical support and daily maintenance. On this basis, the work efficiency of the operator can be improved, the working time can be saved, and the work of the operator can be greatly facilitated. Therefore, the following functions of IoT data processing and analysis software are indispensable.

As shown in Table 2, the functional requirements of the IoT data analysis and processing software include reading background data, data graphing, data processing, data analysis, and data deletion. The software management functions include main window management, software running status management, help, and close the software. The most important functions are data graphing, data processing, and data analysis. The data charting function allows users to distinguish the key data values in the table according to color labels. At the same time, users can flexibly and autonomously choose to draw different variable graphs. By zooming in, zooming out, dragging left and right, and capturing each data point value of the graph, graphic saving, graphic printing, display, hiding of each focus value of the graphic, and so forth are convenient for users to analyze and process the data more deeply. The data processing function allows users to use the data filtering function to filter key data from a large amount of redundant data to filter out effective data points and also use data statistics to classify key data statistically. When the user has limited time or unexpected circumstances such as emergencies, the data cannot be successfully processed and the software can save the data in a new file or picture, which provides a basis for subsequent processing and analysis, which saves the user's time and improves work efficiency. The data analysis function can draw a clear conclusion for the user through data processing and analysis. The above functional requirements are all key steps in the software, and none of the missing steps can constitute the software requirements for data analysis and processing.

3.3. Software Overall Design. The previous question mainly discusses the theoretical knowledge and demand analysis of data analysis based on virtual technology. The content of this chapter will give a detailed description of the design and implementation methods of the Internet of Things data analysis software on the basis of the above research.

3.3.1. Software Overall Design. In view of the functional requirements of the Internet of Things data analysis and processing software, as well as the nonfunctional requirements of the Internet of Things analysis software based on virtual technology, the main functional modules decided to develop are data capture, data filtering, data graphing, protocol analysis, and display. Four modules are stored. The system function is shown in Figure 3.

As shown in Figure 3, there are five main functions of the software design. The data capture function delivers the captured data to the upper application for analysis and processing. Under the environmental conditions of the computer system or mobile phone operating system, carefully analyze the advantages and disadvantages of the data by comparing the three methods of data capture. In order to improve its packet capture performance, it has improved its user cache and system cache and increased the amount of cache so that packet loss is less likely to occur when the network data traffic is too large. Data filtering is to filter out the data packets that have been captured to the data that is useful to the user. In the driving filtering mechanism, all data flowing through is stored in the cache. Data filtering needs to pick out the captured data and copy it to the system cache. This process requires setting filtering rules in advance. Therefore, both the data capture module and the analysis module use the data filtering module, and the whole process greatly improves the system efficiency. The protocol analysis function includes two parts, one is real-time flow monitoring, and the other is comparison of historical data. Real-time flow monitoring is mainly configured according to the characteristics of the data flow, and the specific data flow is filtered in real time. Historical data comparison is to store the captured data through the traffic database, take advantage of the long storage time, and compare them at different time periods to find their differences and solve the problems. The display storage function cannot be ignored. The display storage is directly oriented to the user, and you can imagine its importance in each module. The detailed information of the data packet, such as the protocol, source address, and source port number, as well as the analysis result, will be displayed on the user interface. It can be convenient for users to control and can better display the content to meet the requirements.

Table 2: Functional requirements.

Functional requirements	Functional overview
Read background data	Read and record data from the background. According to the type and structural characteristics of the data, the software automatically processes and analyzes the data and loads the corresponding functional modules.
Data charting	After the data is analyzed, it is displayed in tables and images.
Data processing	The data is screened out for valid data, and key data is established for statistical classification.
Data analysis	Analyze and process data, and give meaningful and valuable conclusions.
Data deletion	Data can be deleted to save memory.
Main window management	Use the software clearly and reasonably.
Software operation status management	Know the activity status of software functions.
Help	Contains basic information such as software information, operating environment, and ownership of the software.
Close the software	Close the software; the program stops running.

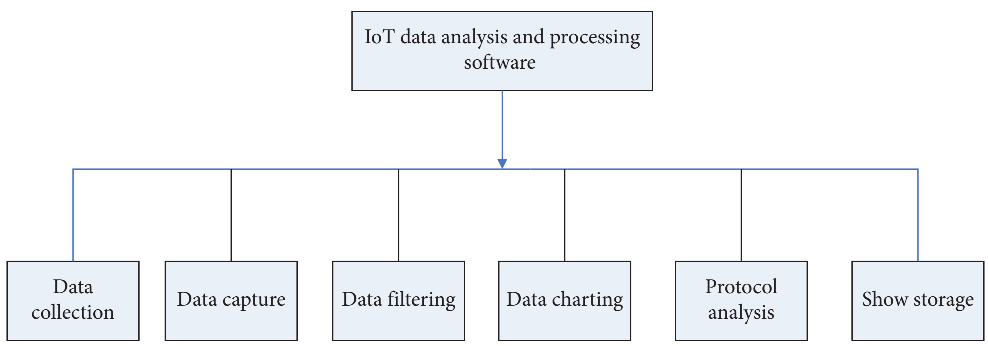

Figure 3: Software function.

3.4. The Software Design of the Data Acquisition Module of the Internet of Things and Data Acquisition Interference

3.4.1. Software Design of Data Acquisition Module. The data acquisition system uses two components with virtual technology, namely, A and B. The two components independently complete the collection and control of the data of their functional modules. B sends the collected data to A through the channel, and finally all the collected data is packaged and uploaded to the IoT data processing system through A according to the protocol. The data acquisition block diagram of the system is shown in Figure 4.

As shown in Figure 4, the design of the data acquisition block diagram is as above, so that the virtual reality data analysis technology is reasonably used to achieve the overall effect of data acquisition. Component B collects pulse signals, digital quantities, and packaged data by polling, opens the serial port to interrupt, receives the extended three-way data, and finally encapsulates all data and sends it. In addition to uploading the collected data, component B also receives data and instructions issued by data processing through serial port interruption. At the same time, component A collects three analog signals through polling mode and receives communication interface data through serial port interruption. Finally, the data is encapsulated according to the protocol and sent to the component. Such data collection will not cause data omissions. The programming process is shown in Figure 5.

3.4.2. Interference of Data Collection. In the application field, the operating environment of the software is complex. When running the software, it is easy to encounter lightning strikes, and when the main power system of the surrounding large-scale equipment is switched and the system fails, the transient voltage surge generated is a great test for the reliable and stable operation of the software. In the data acquisition module, there are many analog acquisition circuit components and long transmission lines. In a complex operating environment, it is easy to introduce interference during signal transmission. Moreover, the accuracy of the collected data is high, and small external interference will also cause large fluctuations in the data. Therefore, in order to ensure the stable and reliable signal acquisition and transmission, the reliability design of the circuit and software operation is particularly important. Among them, the surge interference to the analog acquisition circuit is the most frequent. Therefore, finding the source of interference and analyzing its interference path are a prerequisite for electromagnetic compatibility design.

As shown in Figure 6, when loop 1 works abnormally, it will cause a change in the potential at the top of the common impedance, which will cause interference to loop 2 and affect its normal operation. Common impedance coupling often exists in the grounding system, so pay attention to the grounding treatment of the system. The interference signal generated on the system by the power supply tested in the data acquisition software is transmitted through the

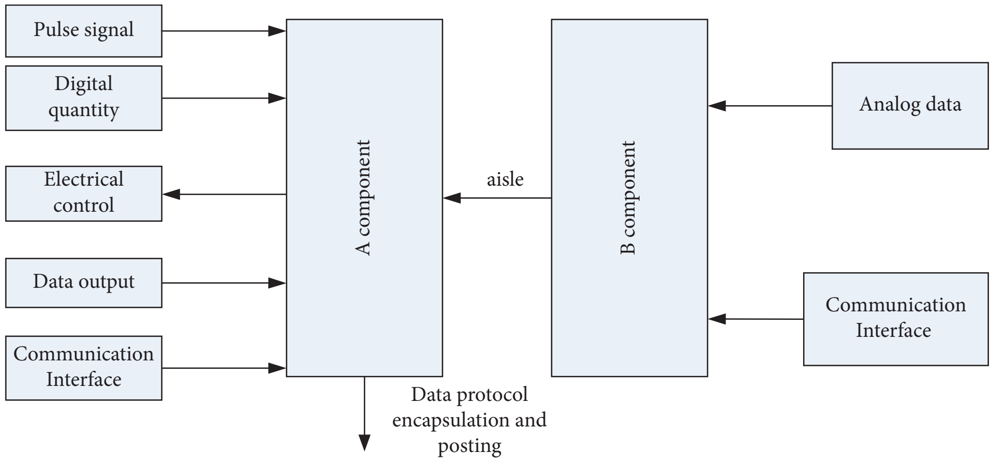

Figure 4: Block diagram of data acquisition.

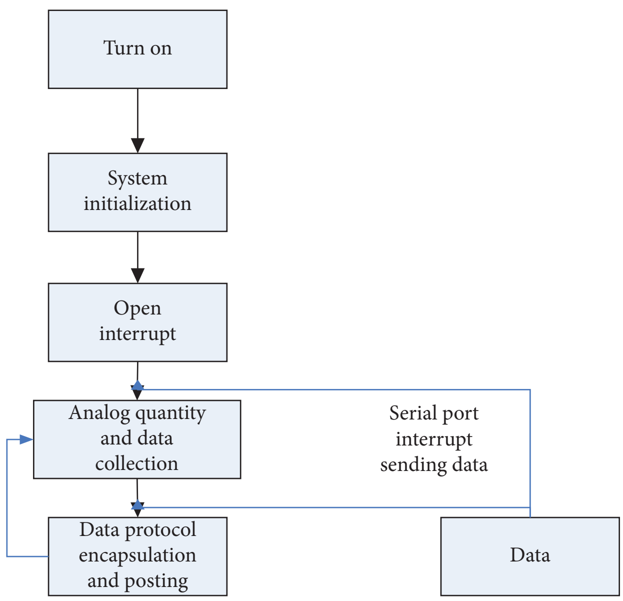

Figure 5: Flow chart of data acquisition program design.

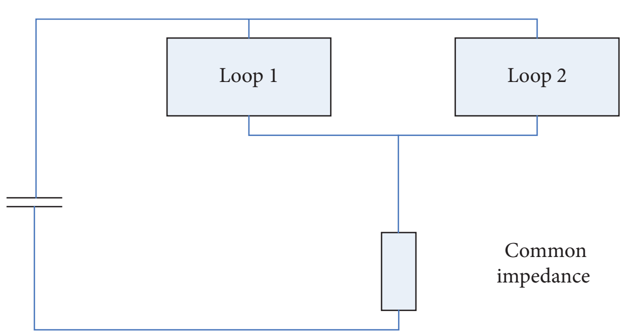

Figure 6: Common impedance coupling loop.

conductive coupling of the wire. The most basic way to suppress this type of interference is to reduce the resistance of the common impedance.

Capacitive coupling is a kind of electrostatic induction. Changes in capacitive coupling can lead to an increase in impedance. For example, there are interference source circuit x and interfered circuit y, and the load of circuit y is RL. There is distributed capacitance z between the adjacent wires x and y of the two circuits. Circuit x is obtained by the voltage division formula. The formula for the induced voltage coupled to circuit y is

$$U_y = \frac{jCR}{1 + jCR} U. \quad (8)$$

When jCR is much smaller than 1, the above formula can be simplified to

$$U_y = jCRU. \quad (9)$$

It can be seen from the above formula that the simplest method is to reduce the distributed capacitance when increasing the distance; shielding and grounding are also common methods. The mutual inductance M between circuits is defined as the magnetic flux generated by the current in circuit x in circuit y. According to Faraday's law, assuming that the magnetic flux density of the magnetic field is B, the induced voltage generated in the closed loop of area S is as shown in the following formula:

$$U = \frac{d}{d_i} \int Y \cdot \mathrm{ds}. \quad (10)$$

The magnetic flux density Y is a vector, which changes sinusoidally with time in a closed loop, and the induced voltage generated is simplified to the following formula:

$$U = jwYs \cos \theta. \quad (11)$$

Among them, the angle is the angle between the vectors Y and S, and the whole is the total coupled magnetic flux. Combined with the definition of mutual inductance M and formula (11), the mutual inductance voltage of the two circuits can be expressed as in the two following formulas:

$$U = jwMI, \tag{12}$$

$$U = M\frac{\mathrm{d}i}{\mathrm{d}t}. \tag{13}$$

Equation (13) describes the basic equation of the inductive coupling between two circuits. Radiation coupling is different from the transmission of conducted interference coupling through channels such as wires. The former refers to the way the interference source couples interference to the victim device in the form of space propagation. Therefore, radiated interference is more difficult to detect, and the reduction of radiated interference mainly starts from its coupling method. The anti-interference calculation is mainly aimed at the interference caused by lightning strikes, transient voltage, and other factors, as well as the interference coupled by the transmission wire when the software is working normally. The electromagnetic compatibility analysis of the system is carried out, and the software reliability is designed and improved to achieve the protection of the system and the improvement.

3.5. Data Capture Function Design. The initialization process is indispensable every time the data capture program runs. It mainly stores the received data packets and then adjusts the network card to a promiscuous mode to maximize the use of the Internet of Things network and wait for the work to proceed.

The system capture data collection method is based on the promiscuous mode, because the Internet of Things has a shared broadcast mechanism. After the network card is set to the promiscuous mode, all data on the network is transmitted to the network interface, and then the data capture module can start working. All the transmitted data packets are captured, and then this data is passed to the extensible application protocol analysis module and display saving module through analysis and filtering, so that it can be used. The existing LAN network transmission methods basically implement the Ethernet standard, so setting the network card in the promiscuous mode is one of the important issues of data capture, and the promiscuous mode network card setting needs to be implemented in the computer system environment through programming, but in the computer system, it is very difficult to program the bottom layer directly in the environment, and it has a good encapsulation of itself, which must be monitored through the development kit, which simplifies programming and improves efficiency. Its realization logic is shown in Figure 7.

The packet filter technology uses advanced programming function library combined with dynamic link library to jointly configure filter rules. Based on this technical principle, the capture library sets the default storage capture data size to 512 k and receives all data flowing through the port through passive monitoring. The sending and receiving of data are based on the computer system grouping. This realization logic is conducive to the realization of the data capture process. The data capture process is shown in Figure 8.

As shown in Figure 8, the data capture process is performed through two capture options. The accuracy of the data can be achieved through two data filtering options selected by the capture: one is whether to start receiving data, and the other is to capture whether it meets the filter conditions. This will not only reduce the data capture function of the Internet of Things but also reduce data packet loss and reduce data loss.

3.6. Data Filtering Module Design. The core of the data filtering module is the setting of filtering rules. It filters and submits the data to the protocol analysis module based on virtual reality-based IoT data information. The implementation process of the data filtering module is shown in Figure 9.

As shown in Figure 9, there are four situations in the design of the data filtering module, namely, the default situation, protocol filtering, host filtering, and network filtering. Its overall idea is as follows: Under normal circumstances, the filtering module does not start, and all data packets flowing through the host are captured and passed to the upper analysis module; different protocols are selected for filtering, and one or several protocols can be filtered; select the targeted host to filter, and filter the source address and destination address of the host; select a specific network segment to filter, and select the same network segment that needs to be researched to achieve filtering. The following filtering methods can be selected and called arbitrarily. According to the above design ideas, the designed network data analysis software should maximize the utilization rate to avoid unnecessary burdens on the system, filter out inconsistent data before operating data capture, and combine the methods of protocol filtering and service filtering.

3.7. Protocol Analysis Module Design. The design of the protocol analysis module is mainly divided into two points: one is the judgment of the data packet type, and the other is the extraction of characteristic parameters. After the above two tasks are completed, the data is delivered to the display storage module to display and store the detailed data content for the user. This part of the content is very important, so the construction of the agreement framework requires comprehensive consideration.

As shown in Figure 10, all captured data packets must parse the header information step by step from the data link layer, network layer, and application layer. First, after the data packet is captured, it is judged whether it is a known protocol, and if it is, it is delivered to the basic application protocol analysis module for analysis; if not, it is directly delivered to the registration module to complete the registration process and then forwarded to the protocol extension chain processing module process, and finally realize the complete analysis of the agreement to get the result. The specific process of data packet analysis is that as long as there is data in the receiving buffer, the analysis thread will be started to analyze the data in it, but at the same time the capture thread must also be started to avoid packet loss. There are certain conditions for the data analysis function;

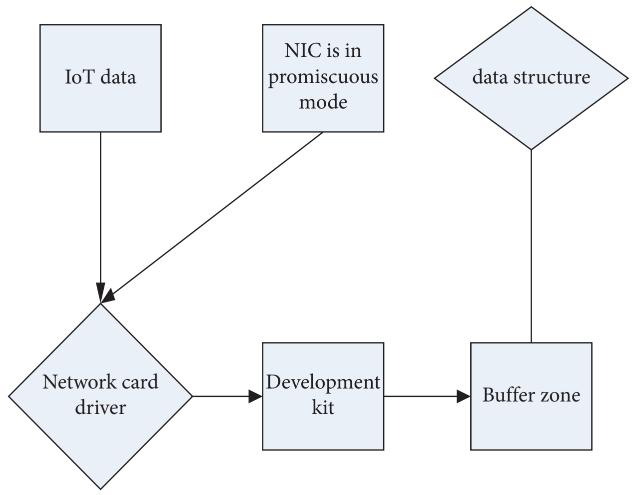

Figure 7: Implementation logic.

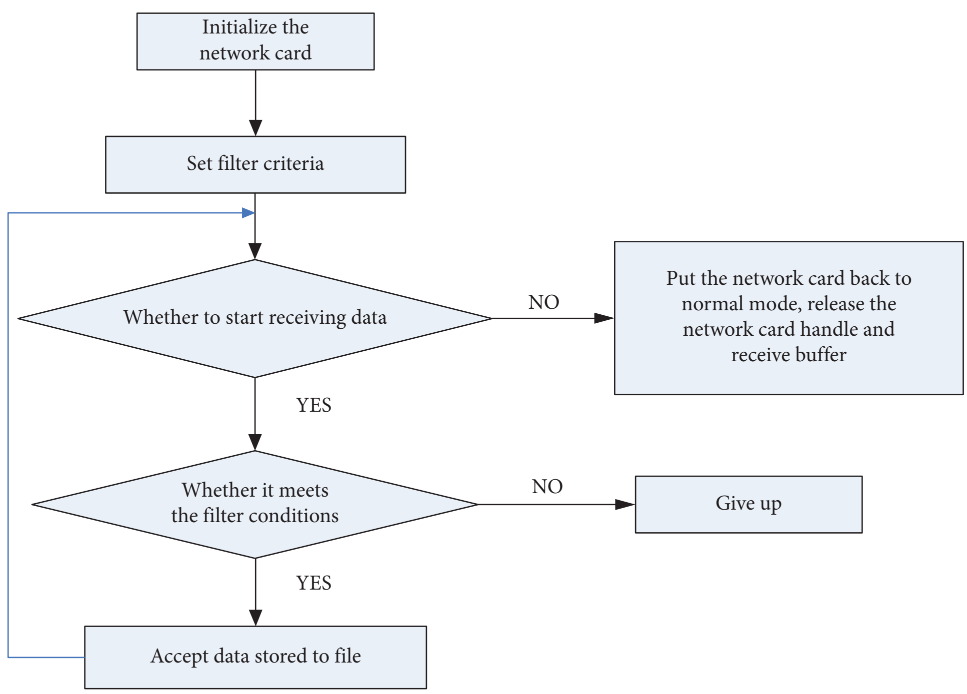

Figure 8: Data capture.

then this condition is that the receiving buffer must have data. In fact, this is an endless loop process. Split the data packet by splitting the frame format of the data packet, and then first set the value in it to 0. When the loop starts, if the value is less than or equal to the data length, then it enters the loop body. When the loop is completed, set the length of the original data packet to determine the value, and finally analyze all the data packets in the buffer. Such analysis and processing can accurately analyze the data and obtain accurate conclusions.

As shown in Figure 11, the protocol analysis flow chart is used to perform step-by-step data protocol analysis through network layer protocols, transport layer protocols, application layer protocols, and other common protocols. By matching from bottom to top layer by layer, most of the captured original data packets are IoT data frames. First, determine the header format of the data link layer frame of the data packet, and then look at its network layer structure and analyze it. ARP is a transport layer protocol or another protocol. If it is a transport layer protocol, then look at its transport layer and determine whether it is UDP or TCP. Use this process to narrow the search range until the application layer gets the desired data.

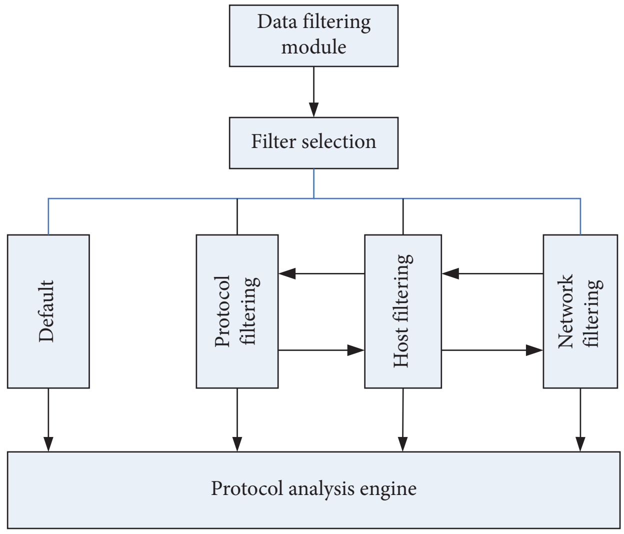

Figure 9: Filtering process method.

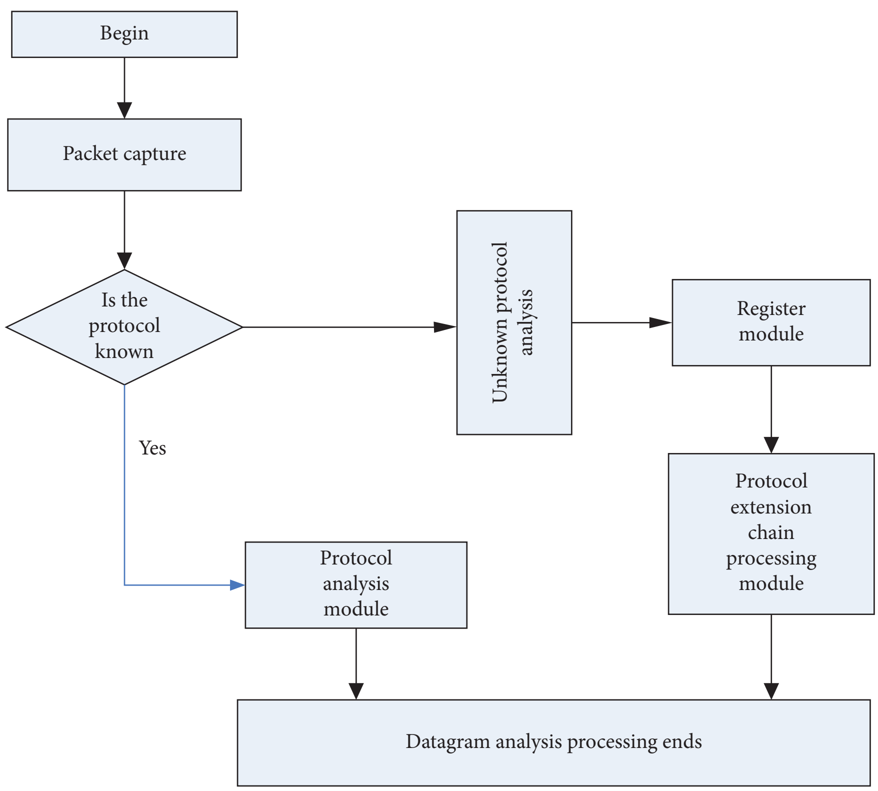

Figure 10: Data protocol analysis process.

4. Software Operating Environment and Testing

4.1. Test Environment

4.1.1. Hardware Environment. As shown in Table 3, the hardware environment mainly includes hardware configuration and operating system. The above-mentioned hardware environment is more able to realize the operation of the data analysis and processing software of the Internet of Things.

4.1.2. Software Environment. As shown in Table 4, the software environment mainly includes operating system, database software, application software, and stress testing software.

4.2. Software Testing

4.2.1. Data Acquisition Interference Test. As shown in Figure 12, under different conditions of suppression improvement, the increase in the acquisition rate is inversely

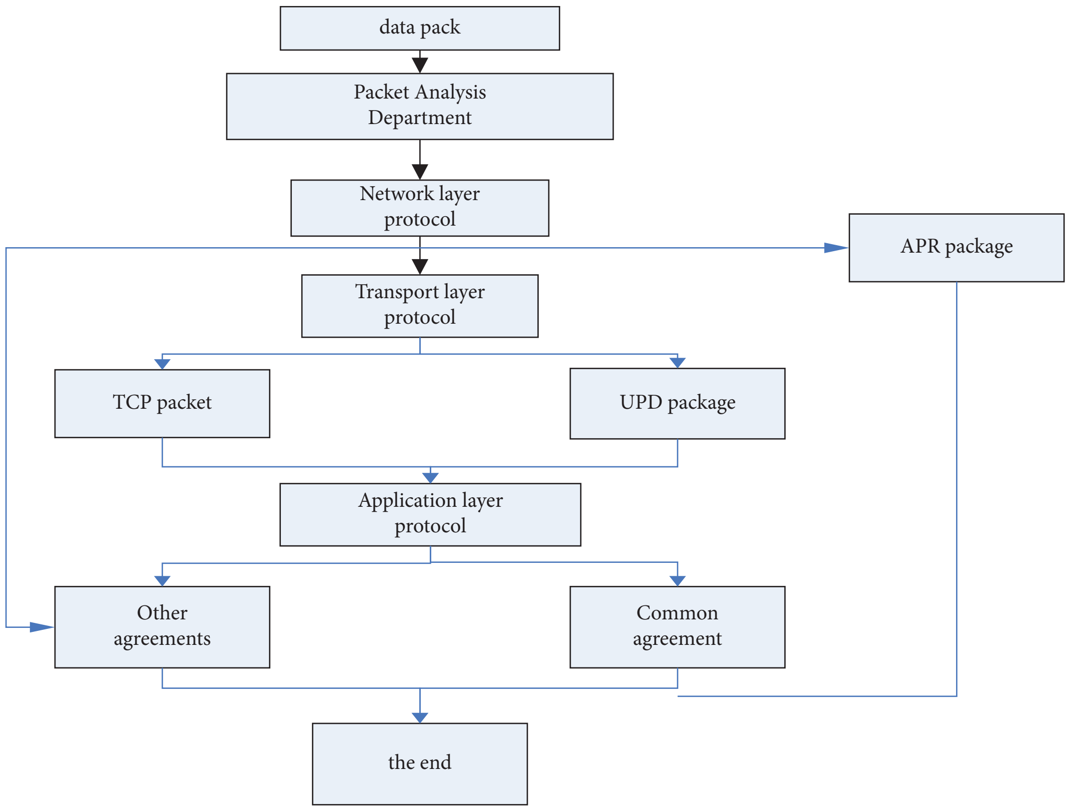

Figure 11: Protocol analysis flow chart.

Table 3: Test hardware environment.

	Client	Database	Server
Hardware configuration	CPU: Intel (R) Core (TM) i3-2120 3.30 GHz Memory: 4 G Hard disk: 500 G	CPU: Intel (R) Xeon (R) E5606 @ 2.13 GHz Memory: 16 G Hard disk: 1 T	CPU: Intel (R) Xeon (R) E3-1220 V2 @ 3.10 GHz Memory: 16 G Hard disk: 500 G
Operating system environment	Microsoft Windows XP	Ubuntu-12.04.1-server-amd64	Ubuntu-12.04.1-server-amd64

Table 4: Test software environment.

Types	Name
Operating system	Microsoft Windows 7
Database software	Oracle 11g
Application	IoT data analysis and processing software
Stress testing software	LoadRunner 11.0

proportional to the interference rate, and the decrease in the interference rate will lead to an increase in the acquisition rate. It shows that the anti-interference design improves the collection rate and the reliability of the software . This verifies the validity and feasibility of the software's data collection function.

4.2.2. Data Capture Test. The data capture test is carried out for the developed IoT data analysis and processing software to test and evaluate the data capture function of the software module. Before the start of the test experiment, keep the test environment the same, with the client models running the software being the same, and ensure that all other conditions are equal. Start the software to download and store data packets to test the capture function and operating environment of the software.

As shown in Table 5, it is found that the size of the packet loss rate is related to the core state buffer and the user state buffer, their relationship is inversely proportional, and it also has a great impact on the performance of the software data capture implementation. It can be seen that when the core buffer and the user buffer increase, the packet loss rate is greatly reduced, the time to implement this function is also

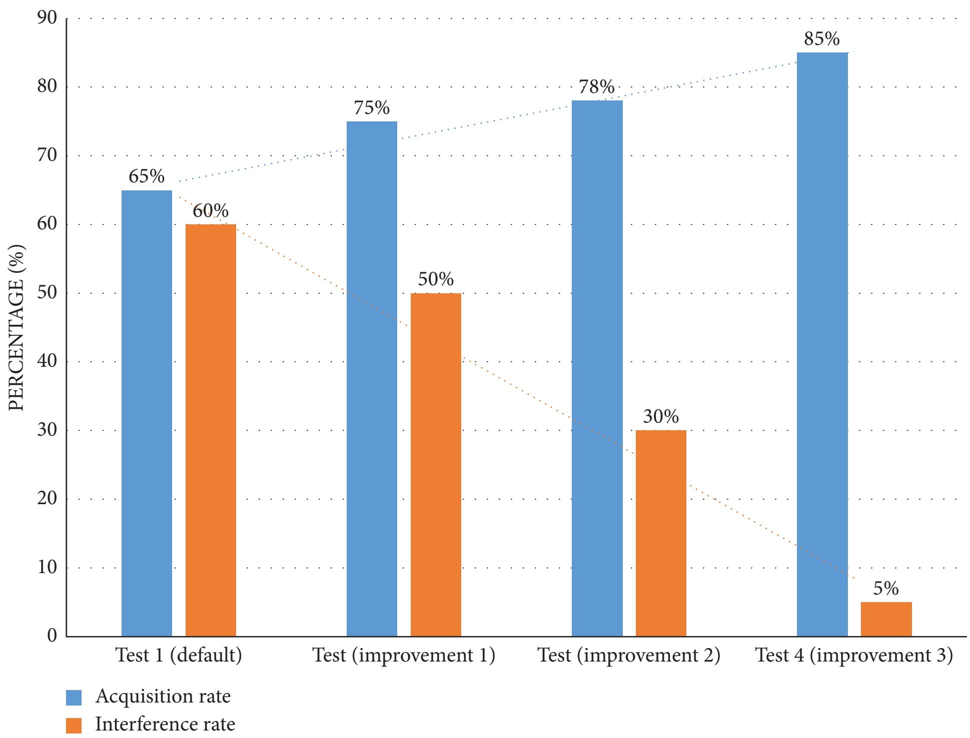

Figure 12: Data acquisition interference test results.

Table 5: Test result table.

Frequency	Download file size (MB)	Core buffer	Average rate	Time	User mode buffer	Packet loss rate
Test 1	20	Default	1000	28	Default	80%
Test 2	20	Default	1100	26	500 KB	70%
Test 3	20	5 MB	1150	25	800 KB	60%
Test 4	20	5 MB	1200	23	1100 KB	40%
Test 5	20	10 MB	1370	20	1300 KB	18%
Test 6	20	10 MB	1410	18	2500 KB	0

Table 6: Test cases.

Numbering	Project	Part of the external conditions	Result
1	Data filtering	After data collection and data capture	In line with expectations
2	Data charting	The data enters the run successfully, and the graph is drawn	In line with expectations
3	Data protocol analysis	Compliant with protocol data	In line with expectations
4	Data protocol processing	Data analysis and processing of different types of protocols	In line with expectations
5	Get conclusion	Show graphs and conclusions	In line with expectations
6	Display data storage	Save diagrams and conclusions	In line with expectations

greatly reduced, and the rate is gradually increased. Because in the Internet of Things there is a large amount of data transmission and sending, this requires us to increase the user buffer area to improve software application performance. The advantage of this approach is to enhance the software system's ability to respond to fast network speeds or increased burst network traffic. The disadvantage is that the cache storage is too much occupied. After testing, the core buffer size was adjusted to 20 MB, and the user buffer size was 2500 KB. Through this adjustment, it was found that the software's packet loss rate was greatly reduced and almost no packet loss occurred.

4.2.3. Data Analysis and Processing Function Test. As shown in Table 6, the functional module tests of the IoT data analysis and processing software are in line with expectations. Passing the functional tests shows the reliability and practicability of the software and also shows that the software basically meets the virtual reality-based IoT data analysis and processing.

5. Discussion

This paper designs the software for data analysis and processing of the Internet of Things. Through the use of data analysis algorithms of virtual reality technology, the analysis of the functional requirements and nonfunctional requirements of the Internet of Things data analysis, the functional modules of the Internet of Things data analysis and processing software are carried out. The software is designed and then the functional modules of the software are tested to achieve the purpose of the software design and to achieve the accuracy of data analysis and processing and data integrity. The lack of research lies in the in-depth understanding of virtual reality technology, but it also provides a certain research foundation for data analysis software and adds some momentum to the development of data analysis software.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by a fund project which is about the application and research of big data precision marketing in the construction of smart campus, which belongs to the Key Scientific Research Platforms and Scientific Research Projects of General Universities in Guangdong Province (Project no. 2018 KTSCX272).

References

[1] A. Faisal, "Computer science: visionary of virtual reality," *Nature*, vol. 551, no. 7680, pp. 298-299, 2017.
[2] J. Dascal, M. Reid, and W. W. Ishak, "Virtual reality and medical inpatients: a systematic review of randomized, controlled trials," *Innovations in Clinical Neuroscience*, vol. 14, no. 1-2, pp. 14–21, 2017.
[3] E. Ronchi, D. Nilsson, S. Kojić et al., "A virtual reality experiment on flashing lights at emergency exit portals for road tunnel evacuation," *Fire Technology*, vol. 52, no. 3, pp. 623–647, 2016.
[4] S. R. Sharar, A. Alamdari, C. Hoffer, H. G. Hoffman, M. P. Jensen, and D. R. Patterson, "Circumplex model of affect: a measure of pleasure and arousal during virtual reality distraction analgesia," *Games for Health Journal*, vol. 5, no. 3, pp. 197–202, 2016.
[5] A. G. Lawson, A. D. Salanitri, and B. B. Waterfield, "Future directions for the development of virtual reality within an automotive manufacturer," *Applied Ergonomics*, vol. 53, no. s 1–2, pp. 323–330, 2016.
[6] J. Munafo, M. Diedrick, and T. A. Stoffregen, "The virtual reality head-mounted display Oculus Rift induces motion sickness and is sexist in its effects," *Experimental Brain Research*, vol. 235, no. 3, pp. 889–901, 2017.
[7] A. M. Ibrahim and C. Karthikeyan, "Predictive machine learning and data acquisition for power quality improvement in facts devices with optimum power flow control based on cross difference progression and coordination examining algorithm," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 18, no. 01, pp. 683–691, 2020.
[8] A. Vankipuram, P. Khanal, and A. Ashby, "Design and development of a virtual reality simulator for advanced cardiac life support training," *IEEE Journal of Biomedical & Health Informatics*, vol. 18, no. 4, pp. 1478–1484, 2017.
[9] J. Gutiérrez-Maldonado, B. K. Wiederhold, and G. Riva, "Future directions: how virtual reality can further improve the assessment and treatment of eating disorders and obesity," *Cyberpsychology, Behavior, and Social Networking*, vol. 19, no. 2, pp. 148–153, 2016.
[10] X. Yao, J. Wang, M. Shen, H. Kong, and H. Ning, "An improved clustering algorithm and its application in IoT data analysis," *Computer Networks*, vol. 159, pp. 63–72, 2019.
[11] G. Suzanne, "IIoT requires data analysis," *Control Engineering: Covering control, instrumentation, and automation systems worldwide*, vol. 63, no. 3, p. 33, 2016.
[12] M. Palaniswami, A. S. Rao, D. Kumar, P. Rathore, and S. Rajasegarar, "The role of visual assessment of clusters for big data analysis: from real-world Internet of things," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 6, no. 4, pp. 45–53, 2020.
[13] G.-H. Jo, S.-B. Jeon, H. Chung, and Y. J. Song, "Sensor data analysis and visualization of IoT system for combat helmet," *Advanced Science Letters*, vol. 23, no. 10, pp. 10342–10345, 2017.
[14] M. Stolpe, "The Internet of things," *ACM SIGKDD Explorations Newsletter*, vol. 18, no. 1, pp. 15–34, 2016.
[15] M. Nardelli, S. Nastic, S. Dustdar, M. Villari, and R. Ranjan, "Osmotic flow: osmotic computing + IoT workflow," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 68–75, 2017.
[16] K. A. Meerja, P. V. Naidu, and S. R. K. Kalva, "Price versus performance of big data analysis for cloud based Internet of things networks," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 1078–1094, 2019.
[17] S. Gill, "Properties data," *Materials Engineering*, vol. 63, no. 3, pp. 33–41, 2016.
[18] I. Omoronyia, U. Etuk, and P. Inglis, "A privacy awareness system for software design," *International Journal of Software Engineering and Knowledge Engineering*, vol. 29, no. 10, pp. 1557–1604, 2019.
[19] A. Dubey, P. Tzeferacos, and D. Q. Lamb, "The dividends of investing in computational software design: a case study," *International Journal of High Performance Computing Applications*, vol. 33, no. 2, pp. 322–331, 2019.
[20] L. Aladib and S. P. Lee, "Pattern detection and design rationale traceability: an integrated approach to software design quality," *IET Software*, vol. 13, no. 4, pp. 249–259, 2019.
[21] C. Rennick, C. C. W. Hulls, and K. N. Mckay, "Introductory engineering decision-making: guiding first-year students to relativism in software design," *IEEE Transactions on Education*, vol. 62, no. 3, pp. 199–208, 2019.
[22] I. Ozkaya, "Ethics is a software design concern," *IEEE Software*, vol. 36, no. 3, pp. 4–8, 2019.
[23] B. Shishkov, "Business modeling and software design," *Lecture Notes in Business Information Processing*, vol. 319, pp. 280–288, 2018.
[24] Akashdeep, Kaur, and Satwinder, "Detecting software bad smells from software design patterns using machine learning algorithms," *International Journal of Applied Engineering Research*, vol. 13, pp. 10005–10010, 2018.
[25] J. Weston, "Software design," *Springer Theses*, vol. 4, pp. 51–66, 2017.
[26] P. Čekan, K. Balog, and J. Harangozó, "Software design to calculate of vibration for a more effective assessment of the safety of the working environment," *Transactions of the Vb*

– *Technical University of Ostrava Safety Engineering*, vol. 11, no. 2, pp. 1–5, 2017.
[27] B. Willhoft and R. Willhoft, “Decoding software design,” *Computing in Science & Engineering*, vol. 19, no. 3, pp. 86-87, 2017.
[28] H. Gomaa, “Real-time software design for embedded systems,” *Microwave Oven Control System Case Study*, vol. 19, pp. 371–416, 2016.
[29] H. Gomaa, “Real-time software design for embedded systems,” *Railroad Crossing Control System Case Study*, vol. 20, pp. 417–450, 2016.
[30] H. Hamidi and M. Jahanshahifard, “The role of the Internet of things in the improvement and expansion of business,” *Journal of Organizational and End User Computing*, vol. 30, pp. 24–44, 201.
[31] Z. Lv and K. Amit, “Big data analysis of Internet of things system,” *ACM Transactions on Internet Technology (TOIT)*, Springer, Berlin, Germany, 2020.
[32] S. K. Biswas, D. Devi, and M. Chakraborty, “A hybrid case based reasoning model for classification in Internet of things (iot) environment,” *Journal of Organizational and End User Computing*, vol. 30, no. 4, pp. 104–122, 2018.
[33] D. Guido, H. Song, and A. Schmeink, *Big Data Analytics for Cyber-Physical Systems: Machine Learning for the Internet of Things*, pp. 1–360, Elsevier, Amsterdam, Netherlands, 2019.
[34] Z. Lv, R. Lou, and J. Li, “Big data analytics for 6G-enabled massive Internet of things,” *IEEE Internet of Things Journal*, vol. 99, p. 1, 2021.