A Real-Time Framework for Structural Health Monitoring Based on the Internet of Things—An Experimental Study

ISHNA JAIN, AJAY KUMAR SREERAMA,
PRAFULLA KALAPATAPU
and VENKATA DILIP KUMAR PASUPULETI

ABSTRACT

Structural Health Monitoring (SHM) is a continuous and autonomous monitoring technique for ensuring structural safety, integrity, and performance without affecting the structure itself. Advances in technology help in identifying structural defects early on, which aids in making decisions pertaining to when repair and rehabilitation are required for civil structures resulting in saving the economy. The Internet of Things (IoT) and cloud services paradigm extends a faster and more reliable SHM strategy. There have been few studies, especially in data driven SHM for a while, where SHM with IoT was implemented over traditional monitoring techniques. However, solutions presented in the recent literature have the following drawbacks viz, latency, power consumption, and real-time synchronization. Most approaches fail to meet real-time requirements, resulting in slower data visualization, retrieval, and analysis, which might lead to late warning alarms eventually leading to catastrophic events. The present research study focuses on these problems by building a completely feasible, cost-efficient, and scalable IoT architecture utilizing AWS cloud-based services with real-time constraints. Data acquired from the sensor nodes are transmitted via a communication module following star topology to the fog layer (DAQ) through MQTT to the AWS cloud and is processed through different cloud services and is visualized immediately making it an asynchronous architecture. Also, the paper presents the tradeoffs for choosing sensors, communication protocols, algorithms, and cloud services for an efficient and secure real-time approach with an aim to provide a promising platform for an efficient and proactive SHM system for civil infrastructures. Our architecture has been tested with Commercially Off-The-Shelf (COTS) hardware on the prototype models developed in the lab and the results have been validated for the same. The extension of this research work for real-time applications has also been discussed.

Ishna Jain[1*], Ajay Kumar Sreerama[2], Prafulla Kalapatapu[3], and Venkata Dilip Kumar Pasupuleti[3]
[1]UG Student, [2]Researcher, [3]Faculty
[1,3]Ecole Centrale School of Engineering, Mahindra University, Hyderabad, India
[2]Mantis Infra Solutions Pvt Ltd, Hyderabad, India
ishna20uece109@mahindrauniversity.edu.in, ajay.sreerama@outlook.com,
prafulla.kalapatapu@mahindrauniversity.edu.in, venkata.pasupuleti@mahindrauniversity.edu.in

INTRODUCTION

The fundamental backbone of any modern society lies in its civil infrastructures and structure. Bridges play a vital role in uniting communities and contributing to the economic and cultural development of a country. Structural Health Monitoring (SHM) refers to the continuous monitoring process of infrastructure assets such as building bridges, and other structures, to evaluate their performance and integrity. The need for SHM stems from reasons including ensuring the safety of people who use the structure, facilitating cost-effective proactive maintenance by identifying potential issues early on, contrary to reactive maintenance strategy, hence extending the lifespan of structures. Therefore, SHM is an invaluable tool that can benefit various industries significantly.

One of the key parameters measured in SHM is vibrations, which are indications of the structure's dynamic behavior. By analyzing the vibrational patterns of a structure over time, it is possible to identify natural frequencies and assess the effectiveness of repairs and modifications [1]. Recent advancements in IoT and cloud services have been implemented to facilitate extensive research in SHM for many years as they make it feasible to monitor structures in real-time.

The key components of typical SHM systems include the structure itself, sensors embedded in bridges for data collection, Data Acquisition Systems (DAQ), a mechanism for data transfer and storage, data management, and data interpretation and diagnosis tools for visualization and analyzation collectively enabling in-depth monitoring of a structure in real-time [2].

Latency, arising from potential delays in data acquisition and transmission, poses a significant obstacle to real-time monitoring capabilities and timely decision-making. Additionally, scalability and maintainability issues emerge due to the complex architecture inherent in IoT systems. Moreover, the substantial cost associated with Data Acquisition Systems (DAQs) often limits the feasibility of consistent and long-term monitoring, leaving structures vulnerable to potential damage. To overcome these challenges, this research paper introduces a novel IoT architecture that leverages Commercially Off-The-Shelf (COTS) hardware and harnesses the capabilities of Amazon Web Services (AWS) cloud services. The proposed architecture aims to address the limitations of current SHM solutions by offering a practical, cost-efficient, and scalable approach for implementation in real-world scenarios. By integrating COTS hardware, efficient data transfer protocols, and AWS cloud services, the proposed architecture enables real-time monitoring, comprehensive data management, and advanced analytics, thereby enhancing structural health assessment. This research seeks to foster innovation in the SHM field by promoting open development and encouraging further advancements in the domain.

LITERATURE REVIEW

Past research in the field of SHM has explored the development of IoT architecture for real-time monitoring. Tahat et.al. [3] presents an IoT system for real-time SHM which is deployed using Azure cloud Services which includes MariaDB as a database and a Virtual Machine (VM) for hosting a XAMPP server. Grafana and Orange data mining tools are also used for data visualization and analysis. While this architecture may have its own advantages, potential drawbacks still exist in terms of cost, latency, scalability, maintenance and complexity for practical applications. Since the architecture follows a server-client (request-response) model, all the requests made by the client are processed through a central server resulting in increased network traffic which may lead to high latency as the response time will increase for the individual requests. Adding on to that, the server, VM and associated resources have to run consistently irrespective of their usage, this can lead to a high rise in the cost of the system depending on the cost scheme by Azure resulting in inefficient resource utilization. Also, maintenance and timely updates of dedicated resources are necessary to ensure scalability and compatibility as the architecture relies on a centralized server, any malfunction or crash in the server can affect the reliability of the whole system. Additionally, the implementation of such a system requires adequate training and expertise in the respective areas. Meng et.al. [4] presents a system which utilizes Apache web server as a centralized server and MySQL deployed on Alibaba Cloud with similar drawbacks of latency, scalability, maintainability and inefficient resource utilization leading to high cost. Malik et.al. [5] propose a low-cost IoT system with Thing speak cloud which includes drawbacks of high latency, high intermittent data update interval and a lossy transfer of data which impacts the real-time monitoring and accuracy of the system. Koene et.al. [6] utilizes simple WebSocket communication between the sensor node and the Graphical User Interface (GUI), few disadvantages to consider is the overhead of WebSocket communication i.e. additional information which encapsulates the actual data compared to IoT-oriented protocols can be much higher which can lead to latency as more bandwidth is consumed which affects the actual data rates, also WebSockets follow request-response model, where a connection request has to be made for every connection to the server, making it inefficient while dealing with concurrent connections with increasing sensor nodes and hence, scalability of such a system can be challenging and although it's a connection-oriented protocol which needs to maintain a constantly open connection consuming a lot of power, it doesn't offer any reliable data transfer options. Chang et.al [7] presents an architecture with HTTP server hosted by NODEMCU which utilizes server-to-server communication having additional drawbacks compared to a WebSocket connection with explicit requests for every data transfer, high network overhead and high processing load on the server which can lead to a less reliable latent system with limited scalability. Abdel et.al [8] developed an architecture with TCP/IP with similar drawbacks which include large data overhead, scalability issues and potential latency.

After an extensive review of scholarly articles, several limitations have been found in existing literature. These constraints encompass the system's scalability and maintainability, identifying the appropriate communication protocol that influences data rate and the expenses associated with continuous monitoring. Given the need for practical and feasible implementation of SHM systems, consideration and overcoming of identified drawbacks is important.

STATE OF THE ART

SHM requires a high sampling rate [9] for accurately analyzing the frequency of a

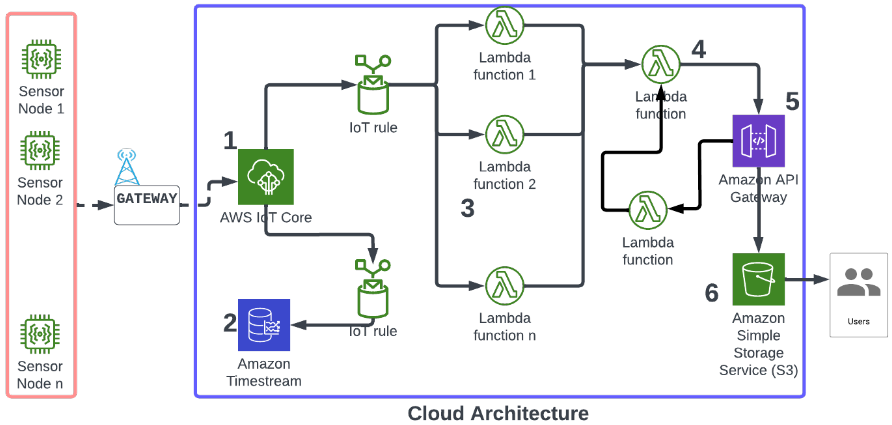

Figure 1. AWS data flow architecture.

structure i.e., the journey of data from sensors to the destination for real-time analysis should support minimal latency and lossless transfer. AWS [10] [11] provides numerous approaches for IoT implementation such as WebSockets, HTTP and REST API through API gateway, TCP/IP connection through VPN or EC2 or MQTT through AWS IoT Core. Drawbacks and challenges associated with these have been discussed with references to the previous research conducted in this area [12].

Figure 1 illustrates the proposed AWS data flow architecture that contributes to the overall research study. The architecture is designed to handle the ingestion, processing, storage, and presentation of data for the research study. It leverages various AWS services to ensure scalability, resilience, and security throughout the data flow.

Data Ingestion

Among all the communication protocols supported by AWS, Message Queue Telemetry Transport (MQTT) protocol [13] [14] is used to communicate from gateway to cloud as it is a lightweight and simple protocol i.e., it has less message overhead and much higher data rate compared to other potential protocols. Less message overhead is preferred since the extra bits encapsulated over raw data can affect the estimated data rate. Also, the protocols supported directly by AWS IoT are preferred keeping in mind their easy and direct implementation. It also supports Quality of Service (QoS) for reliable communication which enables the sender to receive acknowledgment for received data from the receiver and if any failure occurs, data is sent again which ensures lossless transfer. This might add comparatively more delay, but it can be neglected. To keep the architecture simple, QoS is set to 0 indicating that the message is sent at most once with no acknowledgment. Additionally, reliable delivery of messages remains unaffected even with variable connectivity of the network in remote areas as MQTT offers persistent connection throughout ensuring delivery of messages when the network gets back online. As the bidirectional capability of MQTT allows it to transfer data from User Interface (UI) to IoT devices and hence is used for configuring sensor parameters such as data rate, acceleration range, sensitivity and power modes remotely. Also, MQTT supports publish/subscribe paradigm which enables the receiver to subscribe to a topic to get messages which the sender publishes to the topic. Also, the topic name should be kept short and intuitive since it adds to the overhead. Additionally, for secure communication, a private key and a root certificate accessible through the flash memory of the MCU are authenticated.

AWS IoT Core acts as an entry point for the data from the gateway to the AWS cloud (Step 1). IoT rules are used to analyze the MQTT topic stream to which the data is published and based on the SQL query specified in the rule, data is sent to other cloud services for storage and processing.

Data Storage and Persistence

Timestream (Step 2) serves as a primary database and can be accessed through SQL for instantaneous data retrieval and analysis and ensures real-time storage of sensor data. Its dynamic scaling capability makes it scalable and resource-efficient. Compared to other AWS databases like DynamoDB, it has better processing capabilities. Also, the historical and recent data can be analyzed together leveraging the real-time Machine Learning (ML) with AWS Sagemaker.

Data Processing and Analysis

AWS Lambda is utilized as an event-driven server that is activated only in response to a request [15]. SQL query specified in IoT rules is utilized to perform data filtering and categorization based on device ID and send to different Lambda functions (Step 3) facilitating distributed computing. Backend processes such as determining the frequency through Fourier transform and running various algorithms are performed parallelly through Lambda functions and the frequencies obtained through different sensor nodes are compared in Lambda function (Step 4) for damage detection in real-time. Also, it can be used for sending alerts using AWS Simple Notification Service (SNS) in response to significant frequency deviations. Additionally, it ensures that the service operates in a cost-effective manner while still delivering the necessary performance and functionality. This is a serverless and more economical approach compared to traditional server-based solutions like AWS Elastic Compute Cloud (EC2) or physical servers.

Data Presentation and Visualization

The link between the frontend dashboard and backend Lambda function is established with AWS API Gateway (Step 5) which uses WebSocket API for real-time, bidirectional transfer. Frontend for data visualization and sensor configuration console is deployed with AWS S3 (Step 6). Other services such as AWS CloudFront with S3 or AWS Amplify can be utilized for more dynamic applications.

Succinctly, the cloud data flow architecture follows a series of steps for transferring data acquired by sensor nodes through the gateway to the cloud. Firstly, data is published into the destined MQTT topic in AWS IoT core, then through IoT rules is sent to Timestream and respective lambda function where data is processed for every sensor node parallelly. After that, data is received by the backend Lambda function to compare and analyze the multiple frequencies of the structure and send the requested data to WebSocket API which acts as a bridge between the frontend dashboard deployed in

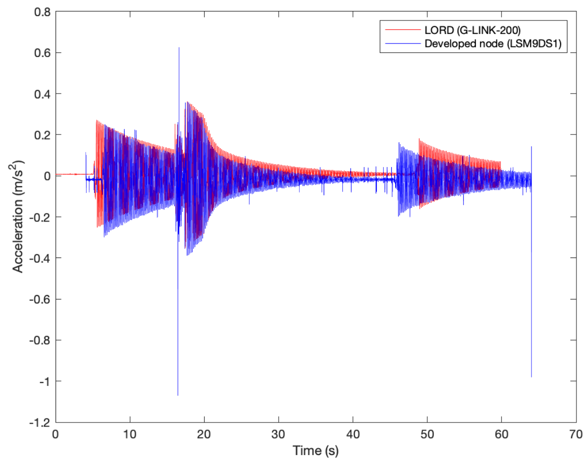

Figure 2. Acceleration value comparison of LORD MicroStrain and LSM9DS1.

AWS S3 and the backend Lambda server. When the deployed website is accessed, a request is sent to the API Gateway through WebSocket API which is then relayed to the backend Lambda function through intermediary Lambda, and the data is accessed asynchronously throughout the connection time.

CASE STUDY

For testing the proposed architecture, sensor nodes and a gateway has been developed to test the latency and accuracy provided by the architecture. Sensor nodes and gateway follow a star topology approach where the gateway acts as a central hub for communication between nodes and the cloud. MicroElectro Mechanical Sensors (MEMS) are used for testing and validation. LORD MicroStrain G-LINK-200 [16] which is a high sensitivity and less noise density sensor with a high sampling rate is compared with a relatively low-priced COTS sensor, LSM9DS1 by ST Microelectronics [17]. Sensor nodes comprise of an accelerometer LSM9DS1 for data acquisition, NodeMCU [18] for data processing and an XBee S2C [19] for wireless transfer from node to gateway through Zigbee protocol. A gateway acts as a bridge between the node and cloud interfaced with XBee S2C for receiving data from sensors, NodeMCU and XBee Cellular LTE [20] for remote transfer to the AWS cloud over MQTT. LSM9DS1 is a triaxial accelerometer with a high Output Data Rate (ODR) and sensitivity which is sufficient to be used as an SHM vibration sensor. Additionally, it supports low-power and high-resolution modes. The experimental setup comprises of a simple cantilever beam affixed to a clamp, with a LORD MicroStrain sensor node and the developed sensor node installed on the opposing surface of the beam. The dimensions of the beam are 85.5 cm x 0.4 cm. The sensors are placed at a 16.5 cm distance from the free end.

Accuracy

LSM9DS1 is configured to match the parameters with +-8g as the input range, the Low pass filter set to 100 Hz and the Output Data Rate (ODR) set to 952 Hz which is

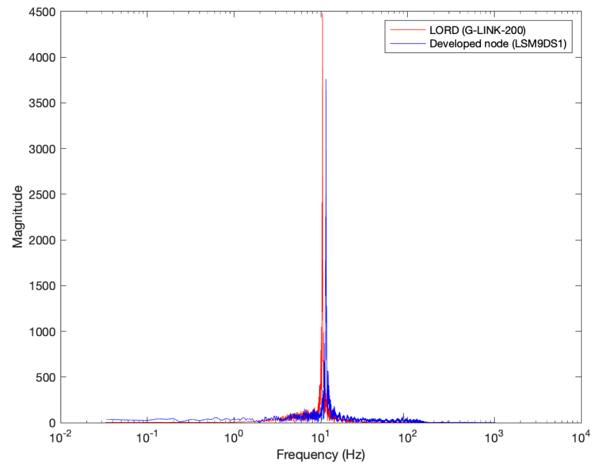

Figure 3. Frequency value comparison of LORD MicroStrain and developed sensor node.

the maximum rate supported by the sensor in high-resolution mode. Since the I2C bus connected to the sensor and SPI bus interfaced with the XBee S2C node supports a maximum data rate of 400 KHz and 4 Mbps respectively and XBee S2C has RF frequency of 2.4 GHz, the ODR becomes a bottleneck.

LORD MicroStrain sensor node is configured with an input range of +-8g, with Low pass filter set to 104 Hz with a sampling rate of 1024 Hz.

The data acquired by the LORD DAQ and proposed DAQ is timestamped and simultaneously stored in CSV files which are then compared in terms of amplitude and frequency.

Figure 2 illustrates the acceleration curve by Lords and developed sensor node. Both measurements almost overlap each other with noticeable spikes in LSM9DS1 values which are due to the high noise density of LSM9DS1 compared to the Lords sensor.

Figure 3 illustrates the frequency obtained by Lords and developed DAQ through Fast Fourier Transform (FFT). Due to the significant differences between their sampling rate and noise density, the percentage error in the frequency domain is 12.88%. With proper calibration and error analysis, the developed node with the proposed architecture can be a cost-optimized alternative for the commercial DAQs used in vibration monitoring in the field of SHM.

Latency

The latency of the whole architecture is calculated by measuring the data with timestamps through serial communication with the sensor node and retrieving the same data stored in Amazon Timestream with timestamps. Latency is calculated by the difference between the timestamp in CSV files obtained from Timestream and serial transfer. For every 100 data points, the mean of latency is calculated and presented in Figure 4.

Figure 4 describes a relationship between latency with an increasing number of data points when monitored continuously. As observed, latency increases from 30 to 60 ms with an increasing number of continuous data points and duration of the connection.

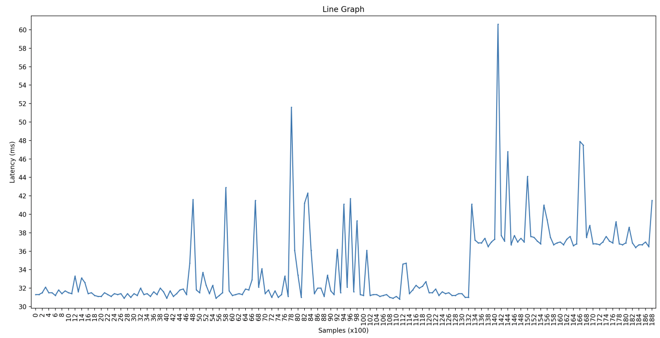

Figure 4. Data points vs Latency graph.

The server-based approaches previously implemented for IoT in SHM require an open connection irrespective of usage, which can lead to an increase in latency with increasing duration. The serverless architecture proposed in this paper eliminates the need for open connection all the time and can be further optimized with selective invoking of the Lambda function.

CONCLUSION

The advancements in IoT and Cloud services SHM has led to innovation and flexibility in the field of SHM. In this paper, a novel scalable and feasible AWS cloud architecture is presented and validated in terms of accuracy and latency taking into consideration the practical and economical implementation. To that end, a sensor node and gateway are developed with COTS hardware and integrated with the proposed architecture. The paper proposes a serverless, asynchronous, and reliable architecture with reduced complexity of implementation, no overhead of managing the underlying infrastructure, improved scalability, and cost optimization.

FUTURE WORK

The proposed architecture provides a foundation for building sophisticated SHM systems. The flexibility provided by AWS allows the architecture for further collaboration with other AWS services. The latency analysis depicts if the duration of monitoring is kept short, the latency will be significantly less compared to longer durations. Taking this into consideration, further optimizations for data acquisition and transmission can be added by leveraging the real-time ML analysis provided by AWS Sagemaker, generating a threshold frequency and only sending the data if a change in threshold is detected. As the architecture is based on services that offer pay according to the use system, cost optimizations can be implemented based on the duration for which these services are used. Also, the development of a sensor node, a protocol for communication between the sensor node, and a gateway for reliably acquiring data from the nodes is still an area with potential research.

REFERENCES

1. Yang, Y., Y. Zhang, and X. Tan. 2021. "Review on Vibration-Based Structural Health Monitoring Techniques and Technical Codes," *Symmetry*, 13(11), ISSN 2073-8994, doi: 10.3390/sym13111998.
2. A, S., J. Regita, B. Rane, and H. H. Lau. 2021. "Structural health monitoring using wireless smart sensor network – An overview," *Mechanical Systems and Signal Processing*, 163, doi: 10.1016/j.ymssp.2021.108113.
3. Tahat, A., A. Al-Zaben, L. S. El-Deen, S. Abbad, and C. Talhi. 2022. "An Evaluation of Machine Learning Algorithms in an Experimental Structural Health Monitoring System Incorporating LoRa IoT Connectivity," in *2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, doi:10.1109/I2MTC48687.2022.9806560.
4. Meng, Q. and S. Zhu. 2020. "Developing IoT Sensing System for Construction-Induced Vibration Monitoring and Impact Assessment," *Sensors*, 20(21), ISSN 1424-8220, doi:10.3390/s20216120.
5. Malik, H., K. Khattak, T. Wiqar, Z. H. Khan, and A. Altamimi. 2019. "Low Cost Internet of Things Platform for Structural Health Monitoring," doi:10.1109/INMIC48123.2019.9022801.
6. Koene, I., V. Klar, and R. Viitala. 2020. "IoT connected device for vibration analysis and measurement," *HardwareX*, 7:e00109, doi:10.1016/j.ohx.2020.e00109.
7. Chang, H.-F. and M. Shokrolah Shirazi. 2021. "Integration with 3D Visualization and IoT-Based Sensors for Real-Time Structural Health Monitoring," *Sensors*, 21:6988, doi:10.3390/s21216988.
8. AbdelRaheem, M., M. Hassan, U. S. Mohammed, and A. A. Nassr. 2022. "Design and implementation of a synchronized IoT-based structural health monitoring system," *Internet of Things*, 20:100639, ISSN 2542-6605, doi:https://doi.org/10.1016/j.iot.2022.100639.
9. Tang, D., J. Chen, W. Wu, L. Jin, Q. Yue, B. Xie, S. Wang, and J. Feng. 2020. "Research on sampling rate selection of sensors in offshore platform shm based on vibration," *Applied Ocean Research*, 101:102192, ISSN 0141-1187, doi:https://doi.org/10.1016/j.apor.2020.102192.
10. Pierleoni, P., R. Concetti, A. Belli, and L. Palma. 2020. "Amazon, Google and Microsoft Solutions for IoT: Architectures and a Performance Comparison," *IEEE Access*, 8:5455–5470, doi:10.1109/ACCESS.2019.2961511.
11. Sääf, A. 2019. "Cloud Services for Building the Modern Internet of Things," .
12. Bayılmış, C., M. A. Ebleme, Ünal Çavuşoğlu, K. Küçük, and A. Sevin. 2022. "A survey on communication protocols and performance evaluations for Internet of Things," *Digital Communications and Networks*, 8(6):1094–1104, ISSN 2352-8648, doi:https://doi.org/10.1016/j.dcan.2022.03.013.
13. Sasaki, Y. and T. Yokotani. 2019. "Performance Evaluation of MQTT as a Communication Protocol for IoT and Prototyping," *Advances in Technology Innovation*, 4(1):21–29.
14. Thangavel, D., X. Ma, A. Valera, H.-X. Tan, and C. Tan. 2014. "Performance evaluation of MQTT and CoAP via a common middleware," doi:10.1109/ISSNIP.2014.6827678.
15. Esposito, M., A. Belli, L. Palma, and P. Pierleoni. 2023. "Design and Implementation of a Framework for Smart Home Automation Based on Cellular IoT, MQTT, and Serverless Functions," *Sensors*, 23:4459, doi:10.3390/s23094459.
16. LORD, "G-LINK-200," https://www.microstrain.com/sites/default/files/g-link-$200_datasheet_8400-0102_rev_h.pdf$.
17. ST Microelectronics, "LSM9DS1," https://www.st.com/resource/en/datasheet/lsm9ds1.pdf.
18. Read the Docs, "NodeMCU," https://nodemcu.readthedocs.io/en/release/.

19. Digi, "XBee S2C," https://www.digi.com/resources/library/data-sheets/ds$_xbee - s2c - 802 - 15 - 4$.
20. Digi, "XBee LTE," https://www.digi.com/resources/documentation/digidocs/PDFs/9000152 5. pdf.
21. Imtiaz Jaya, N. and M. F. Hossain. 2018. "A Prototype Air Flow Control System for Home Automation Using MQTT Over Websocket in AWS IoT Core," in *2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 111–1116, doi:10.1109/CyberC.2018.00032.
22. Scuro, C., P. F. Sciammarella, F. Lamonaca, R. S. Olivito, and D. L. Carni. 2018. "IoT for structural health monitoring," *IEEE Instrumentation Measurement Magazine*, 21(6):4–14, doi:10.1109/MIM.2018.8573586.
23. Lee, J.-L., Y.-Y. Tyan, M.-H. Wen, and Y.-W. Wu. 2017. "Development of an IoT-based bridge safety monitoring system," in *2017 International Conference on Applied System Innovation (ICASI)*, pp. 84–86, doi:10.1109/ICASI.2017.7988352.
24. Abdelgawad, A. and K. Yelamarthi. 2017. "Internet of Things (IoT) Platform for Structure Health Monitoring," *Wireless Communications and Mobile Computing*, 2017:1–10, doi:10.1155/2017/6560797.
25. Wang, J., Y. Fu, and X. Yang. 2017. "An integrated system for building structural health monitoring and early warning based on an Internet of things approach," *International Journal of Distributed Sensor Networks*, 13:2017, doi:10.1177/1550147716689101.
26. Mechitov, K. and W. Kim. 2004. "High-Frequency Distributed Sensing for Structure Monitoring," .
27. Alzughaibi, A., A. Ibrahim, A. Eltawil, Y. na, and S. El-Tawil. 2019. "Post-Disaster Structural Health Monitoring System Using Personal Mobile-Phones," pp. 1–4, doi:10.1109/WISNET.2019.8711805.
28. Muttillo, M., V. Stornelli, R. Alaggio, R. Paolucci, L. Di Battista, T. de Rubeis, and G. Ferri. 2020. "Structural Health Monitoring: An IoT Sensor System for Structural Damage Indicator Evaluation," *Sensors*, 20(17), ISSN 1424-8220, doi:10.3390/s20174908.