OPTIMIZING BIG DATA STORAGE AND COMPRESSION FOR IOT APPLICATIONS USING NETWORK ATTACHED STORAGE

[1] **Basava Ramanjaneyulu Gudivaka**
Genpact India Private Limited, Telangana, India
basava.gudivaka537@gmail.com

[2] **Punitha Palanisamy**
SNS College of Technology,
Coimbatore, Tamil Nadu, India.
Punithapalanisamy93@gmail.com

ABSTRACT

The Internet of Things (IoT) has led to an exponential growth in data generation, creating significant challenges in data storage, processing, and transmission. Existing IoT systems often struggle with managing large-scale data, facing issues such as high storage costs, inefficient data transmission, and slow processing speeds. This paper addresses these challenges by developing an efficient framework that optimizes the management of IoT-generated big data by integrating data compression, and storage techniques. The proposed system begins with data collection, where IoT-generated big data is gathered from various sensors. The next step is data preprocessing, where missing values are handled using median imputation, and outliers are removed using the Interquartile Range method. The preprocessed data then undergoes compression using Zstandard, reducing its size while maintaining integrity. The compressed data is stored in Network Attached Storage, ensuring scalable and efficient storage. Data retrieval is optimized for efficient processing and access. Finally, the system ensures reduced storage requirements and enhanced performance for large-scale IoT data. The results demonstrate a significant reduction in storage utilization, from 100% at 100 MB to 30% at 500 MB, and the compression time increases from 2 seconds at 100 MB to 9 seconds at 500 MB. The contributions of this work lie in the integration of compression, preprocessing, and storage optimization techniques to handle large-scale IoT data more efficiently, ensuring reduced storage costs and improved system performance in IoT applications.

Keywords: Internet of Things (IoT), Big Data, Zstandard, Network Attached Storage and IoT Data Management.

1 INTRODUCTION

The Internet of Things (IoT) has revolutionized data generation by connecting billions of devices worldwide, leading to an unprecedented surge in data volume, velocity, and variety [1]. This proliferation of IoT devices has significantly impacted data management and storage infrastructures, necessitating innovative solutions to handle the massive influx of data [2]. Traditional data processing systems often struggle to efficiently process and store the vast amounts of data generated by IoT devices, leading to challenges in real-time data analytics and decision-making [3]. To address these challenges, integrating advanced data preprocessing and compression techniques with scalable storage solutions like Network Attached Storage (NAS) has become essential. Such integration aims to enhance data handling capabilities, reduce storage requirements, and improve overall system performance in IoT applications [4]. The proposed framework seeks to address these critical issues by combining efficient data preprocessing, compression, and storage strategies tailored for IoT-generated big data.

Existing approaches to managing Internet of Things (IoT) data face significant challenges that hinder their effectiveness [5]. Traditional data management systems often struggle with the vast volumes, high velocity, and diverse variety of data generated by IoT devices, leading to difficulties in real-time processing and analysis. Additionally, the heterogeneity of IoT devices, each utilizing different communication protocols and data formats, complicates data integration and standardization efforts [6]. Security and privacy concerns further exacerbate these challenges, as the distributed nature of IoT systems makes them susceptible to breaches and unauthorized access [7]. Moreover, the lack of scalability in traditional storage solutions poses problems in accommodating the exponential growth of IoT data, necessitating the development of more robust and adaptable data management strategies [8]. Addressing these issues requires innovative approaches that can handle the dynamic and complex nature of IoT data effectively.

The proposed framework overcomes the limitations of existing systems by integrating median imputation, IQR-based outlier removal, Zstandard compression, and NAS-based storage into a cohesive solution. This integration ensures efficient handling of missing data, removal of anomalies, effective data compression, and scalable storage, tailored to the specific needs of IoT applications. The novelty of this study lies in its holistic approach, combining these techniques to create a unified framework that enhances the performance, scalability, and reliability of IoT data management systems.

The organization of the paper is as follows: Section 2 reviews related works. Section 3 describes the methodology. Section 4 presents the experimental setup and evaluation metrics. Finally, Section 5 concludes the paper.

2 LITERATURE SURVEY

Several existing works have addressed the challenges associated with managing IoT big data, focusing on compression, encryption, and data transmission. For example, Xue et al. (2015) proposed a big data dynamic compressive sensing system architecture to optimize data transmission by reducing the amount of data required to be sent [9]. However, while compressive sensing offers data reduction, it often faces limitations in terms of computational complexity and reconstruction errors, especially in real-time applications [10]. Similarly, Jiancheng et al. (2017) developed a parallel algorithm for wireless data compression and encryption, aimed at reducing transmission delays while securing the data [11]. The drawback of this method lies in the potential for high computational overhead during the encryption process, which may impact the efficiency of IoT systems that require quick data processing [12].

Bertino et al. (2015) focused on the development of sensor-based big data cyberinfrastructures to manage large-scale data [13]. Their approach provided scalable and secure systems for data processing but did not fully address the challenges of data redundancy and real-time data handling [14]. Likewise, Zhu et al. (2015) worked on multimedia big data computing techniques, which emphasized the need for efficient storage and retrieval [15]. Despite their contributions, these methods were often not adaptable to IoT systems, where data is continuously generated in real time and requires immediate action [16]. Sadiq Ali Khan et al. (2017) explored big data management in the connected world of IoT, proposing methods for seamless data integration [17]. However, their methods did not adequately address the computational limitations posed by large-scale IoT data processing [18].

These existing methods, although significant, often operate independently and fail to provide a holistic solution for managing IoT big data. They may face challenges with scalability, real-time processing, and integration with diverse IoT environments. The proposed framework overcomes these drawbacks by integrating efficient data preprocessing techniques, advanced compression algorithms like Zstandard (Zstd), and scalable storage systems such as Network Attached Storage (NAS). This unified approach ensures reduced data loss, minimized computational overhead, and optimized

storage, making it more suitable for handling the dynamic and large-scale data generated by IoT applications.

2.1 Problem Statement

Although significant progress has been made in IoT and IIoT data management, several critical challenges still remain unaddressed, particularly in the areas of storage efficiency and reducing transmission costs [19]. The problems include the inability of current systems to handle large-scale data efficiently, high computational overhead, and bandwidth issues that affect overall system performance [20]. The work is proposed to overcome these challenges by integrating advanced compression techniques, efficient data preprocessing, and edge-based analytics, ensuring scalable and cost-effective management of IoT data.

3 METHODOLOGIES

The proposed system starts with data collection, where large volumes of IoT-generated big data are gathered from various sensors deployed across the system. The data then moves to data preprocessing, which handles missing values through median imputation and detects outliers using the Interquartile Range (IQR) method. After preprocessing, the data undergoes compression using Zstandard (Zstd), which reduces the data's size without sacrificing significant information. The compressed data is then organized for optimal storage and accessibility. Finally, all data is stored in Network Attached Storage (NAS), providing a scalable, reliable, and cost-effective solution for storing IoT big data, ensuring efficient data management in IoT applications. This process is illustrated in Figure 1.

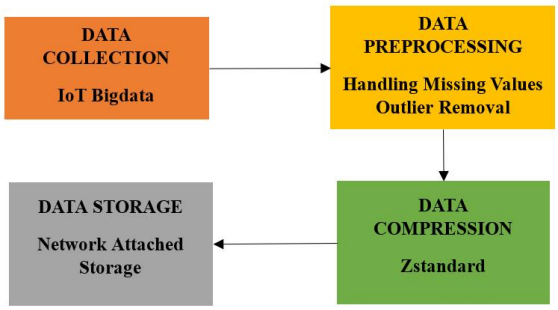

Figure 1: Workflow for IoT Data Management and Optimization

3.1 Data Collection

Data collection involves gathering large volumes of IoT-generated big data from various sensors embedded in devices throughout the system. These sensors continuously monitor different parameters such as temperature, humidity, motion, and more. The data is collected in real-time or at periodic intervals, depending on the application's requirements. The gathered data is then transmitted to a centralized system or storage solution for further processing. Ensuring the accuracy and consistency of this data is crucial, as it serves as the foundation for all subsequent analysis and actions. Effective data collection is vital for managing and optimizing IoT systems, particularly when dealing with large-scale, dynamic environments.

3.2 Data Preprocessing

After the data is collected, the first step in data preprocessing is handling missing values. Median imputation is used to replace missing values with the median of the available data in the dataset. This method is effective as it reduces the impact of outliers and ensures the data distribution is not skewed. By filling in missing values, the dataset becomes more complete for subsequent analysis.

The second step in data preprocessing is removing outliers. The Interquartile Range (IQR) method is applied to detect values that fall outside the acceptable range, calculated by the difference between the first and third quartiles. Any data points beyond 1.5 times the IQR are considered outliers and are removed, ensuring that the dataset reflects more accurate and reliable values.

3.3 Data Compression

After the data has been pre-processed, data compression is applied to reduce the storage requirements and improve transmission efficiency. Zstandard (Zstd) compression is used, a modern algorithm that balances high compression ratios with fast decompression speeds. This method ensures that the size of the pre-processed data is significantly reduced without sacrificing data integrity. Zstd is particularly beneficial for large datasets generated by IoT systems, as it minimizes the storage footprint while maintaining quick access for further analysis. The compressed data is stored efficiently, facilitating faster processing and reduced bandwidth usage during transmission. This compression step is crucial for handling the large volumes of data generated by IoT devices in a scalable and resource-efficient manner.

Zstandard (Zstd) is a lossless data compression algorithm that provides high compression ratios and fast speeds. The compression process can be mathematically represented by applying a compression function C to the original dataset D to generate the compressed dataset D_c is expressed as equation (1),

$$D_c = C(D) \tag{1}$$

Where, D represents the original, uncompressed data. D_c is the compressed data after applying the Zstd compression function C.

Zstd uses a combination of dictionary compression and entropy encoding. It splits the input data into blocks and applies a series of operations to reduce redundancy, including:

A pre-trained dictionary is used to replace repeated sequences of bytes in the input data with shorter representations. The algorithm encodes the data using variable-length codes to represent frequent patterns with shorter bit lengths and infrequent patterns with longer bit lengths.

The decompression function D_d restores the original data from the compressed dataset and its represented as equation (2),

$$D = D_d(D_c) \tag{2}$$

Where, D_c is the compressed data. D is the original uncompressed data restored after decompression. The Zstd algorithm ensures that $D_d(C(D)) = D$, meaning the decompressed data is identical to the original data, preserving data integrity. The mathematical model behind Zstd is designed to optimize for both speed and compression efficiency, making it suitable for large datasets such as those generated by loT systems.

3.4 Data Storage

After the data has been compressed, data storage is the next crucial step. The compressed data is stored in a Network Attached Storage (NAS) system, providing a scalable and efficient solution for handling large volumes of data. NAS ensures centralized storage, allowing multiple devices to access the data over a network. By storing compressed data, NAS reduces the required storage space and minimizes the associated costs. Additionally, NAS provides high availability and data redundancy, ensuring that the compressed data remains accessible and secure. This storage method is particularly beneficial for IoT applications, where data size and retrieval speed are critical factors.

4 RESULTS

The results section presents the performance metrics of the proposed system for managing compressed IoT data. It includes the evaluation of storage utilization and compression time, highlighting the efficiency and computational demands of the Zstandard (Zstd) compression algorithm. The following graphs illustrate the relationship between data size, storage utilization, and compression time. These results demonstrate the effectiveness of the proposed compression technique in optimizing storage and processing efficiency for IoT systems.

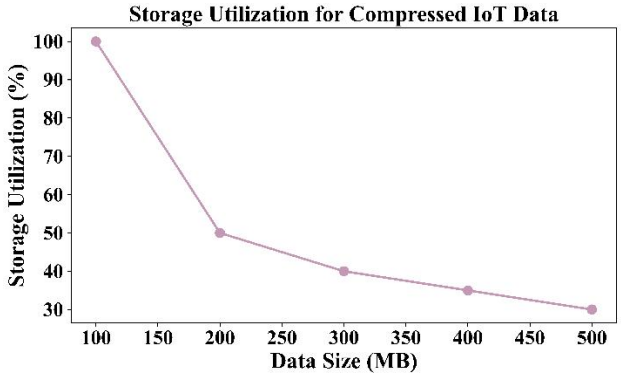

Figure 2: Storage Utilization for compressed IoT data

Figure 2 presents the storage utilization for compressed IoT data. The graph illustrates how storage utilization decreases as the data size increases, reflecting the efficiency of the compression technique. At a data size of 100 MB, the storage utilization is 100%, while at 200 MB, it drops to 50%. As the data size increases further, the storage utilization continues to decrease: at 300 MB it is 40%, at 400 MB it is 35%, and at 500 MB, it reaches 30%. This demonstrates that the compression technique significantly reduces storage requirements, especially for larger datasets, which is vital for efficiently managing big data generated by IoT systems.

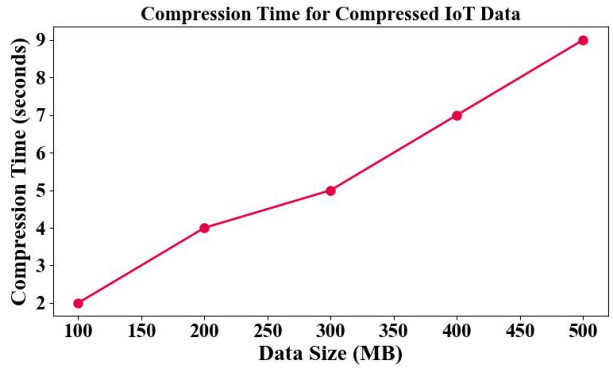

Figure 3: Compression time IoT data

Figure 3 illustrates the compression time for compressed IoT data using the Zstandard (Zstd) compression algorithm. As the data size increases, the compression time also increases, which is expected as larger datasets require more processing time. For data sizes of 100 MB, 200 MB, 300 MB, 400 MB, and 500 MB, the compression times are 2, 4, 5, 7, and 9 seconds, respectively. This trend demonstrates the linear relationship between data size and the time required for compression using Zstd. The graph emphasizes the computational demand for compressing larger IoT datasets, which is an important consideration when optimizing for efficient data processing.

5 CONCLUSIONS

In this work, optimizing the management of IoT-generated big data by integrating data preprocessing, compression, and storage techniques has been achieved. The results show that the compression

method significantly reduced storage utilization, with storage requirements decreasing from 100% at 100 MB to 30% at 500 MB. Additionally, the compression time increased from 2 seconds at 100 MB to 9 seconds at 500 MB, highlighting the relationship between data size and processing time. This integrated approach ensures efficient storage and retrieval of IoT data, significantly reducing storage costs while maintaining data integrity. The system proves to be highly scalable and resource-efficient, making it well-suited for large-scale IoT applications. Future work will focus on enhancing compression algorithms further, exploring real-time data processing capabilities, and optimizing the system for varied IoT environments to improve performance and reduce computational overhead in diverse scenarios.

REFERENCES

[1] M. Li, Y. Liu, and Y. Cai, "A Dynamic Processing System for Sensor Data in IoT," *Int. J. Distrib. Sens. Netw.*, vol. 11, no. 8, p. 750452, Aug. 2015, doi: 10.1155/2015/750452.

[2] F. D'Andria *et al.*, "Data Movement in the Internet of Things Domain," in *Service Oriented and Cloud Computing*, S. Dustdar, F. Leymann, and M. Villari, Eds., Cham: Springer International Publishing, 2015, pp. 243–252. doi: 10.1007/978-3-319-24072-5_17.

[3] R. K. Barik, H. Dubey, A. B. Samaddar, R. D. Gupta, and P. K. Ray, "FogGIS: Fog Computing for geospatial big data analytics," in *2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)*, Dec. 2016, pp. 613–618. doi: 10.1109/UPCON.2016.7894725.

[4] H. Dubey, J. Yang, N. Constant, A. M. Amiri, Q. Yang, and K. Makodiya, "Fog Data: Enhancing Telehealth Big Data Through Fog Computing," in *Proceedings of the ASE BigData & SocialInformatics 2015*, in ASE BD&SI '15. New York, NY, USA: Association for Computing Machinery, Oct. 2015, pp. 1–6. doi: 10.1145/2818869.2818889.

[5] C. L. Chen, A. Mahjoubfar, and B. Jalali, "Optical Data Compression in Time Stretch Imaging," *PLOS ONE*, vol. 10, no. 4, p. e0125106, Apr. 2015, doi: 10.1371/journal.pone.0125106.

[6] Z. Chen *et al.*, "A survey of bitmap index compression algorithms for Big Data," *Tsinghua Sci. Technol.*, vol. 20, no. 1, pp. 100–115, Feb. 2015, doi: 10.1109/TST.2015.7040519.

[7] H. Aly, M. Elmogy, and S. Barakat, "Big Data on Internet of Things: Applications, Architecture, Technologies, Techniques, and Future Directions," *Int. J. Comput. Sci. Eng.*, vol. 4, no. 6, 2015.

[8] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 75–87, Feb. 2017, doi: 10.1109/JIOT.2016.2619369.

[9] J.-W. Xue, X.-K. Xu, F. Zhang, ,School of Management, Northwestern Polytechnical University, Xi'an 710072, and ,School of Information Engineering, Yulin University, Yulin 719000, "Big data dynamic compressive sensing system architecture and optimization algorithm for internet of things," *Discrete Contin. Dyn. Syst. - S*, vol. 8, no. 6, pp. 1401–1414, 2015, doi: 10.3934/dcdss.2015.8.1401.

[10] B. R. Stojkoska and Z. Nikolovski, "Data compression for energy efficient IoT solutions," in *2017 25th Telecommunication Forum (TELFOR)*, Belgrade: IEEE, Nov. 2017, pp. 1–4. doi: 10.1109/TELFOR.2017.8249368.

[11] Q. Jiancheng, L. Yiqin, and Z. Yu, "Parallel Algorithm for Wireless Data Compression and Encryption," *J. Sens.*, vol. 2017, no. 1, p. 4209397, 2017, doi: 10.1155/2017/4209397.

[12] A. P. Plageras *et al.*, "Efficient Large-scale Medical Data (eHealth Big Data) Analytics in Internet of Things," in *2017 IEEE 19th Conference on Business Informatics (CBI)*, Jul. 2017, pp. 21–27. doi: 10.1109/CBI.2017.3.

[13] E. Bertino, S. Nepal, and R. Ranjan, "Building Sensor-Based Big Data Cyberinfrastructures," *IEEE Cloud Comput.*, vol. 2, no. 5, pp. 64–69, Sep. 2015, doi: 10.1109/MCC.2015.106.

[14] Z. Chen and J. Yan, "Fast KNN search for big data with set compression tree and best bin first," in *2016 2nd International Conference on Cloud Computing and Internet of Things (CCIOT)*, Oct. 2016, pp. 97–100. doi: 10.1109/CCIOT.2016.7868311.

[15] W. Zhu, P. Cui, Z. Wang, and G. Hua, "Multimedia Big Data Computing," *IEEE Multimed.*, vol. 22, no. 3, pp. 96-c3, Jul. 2015, doi: 10.1109/MMUL.2015.66.

[16] A.-M. Rahmani *et al.*, "Smart e-Health Gateway: Bringing intelligence to Internet-of-Things based ubiquitous healthcare systems," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Jan. 2015, pp. 826–834. doi: 10.1109/CCNC.2015.7158084.

[17] M. Sadiq Ali Khan *et al.*, "Big Data Management in Connected World of Internet of Things," *Indian J. Sci. Technol.*, vol. 10, no. 29, pp. 1–9, Feb. 2017, doi: 10.17485/ijst/2017/v10i29/117328.

[18] Zhang, Y., He, Q., Xiang, Y., Zhang, L. Y., Liu, B., Chen, J., & Xie, Y. (2017). Low-cost and confidentiality-preserving data acquisition for internet of multimedia things. *IEEE Internet of Things Journal*, *5*(5), 3442-3451.

[19] Sathiya, Aravindhan K., and D. Sathiya. "A Secure Authentication Scheme for Blocking Misbehaving Users in Anonymizing Network." International Journal of Computer Science and Technology 4, no. 1 (2013): 302-304.

[20] R. N. Karthika, C. Valliyammai, and D. Abisha, "Perlustration on techno level classification of deduplication techniques in cloud for big data storage," in *2016 Eighth International Conference on Advanced Computing (ICoAC)*, Jan. 2017, pp. 206–211. doi: 10.1109/ICoAC.2017.7951771.