Journal of
Multimedia Technology &
Recent Advancements

ISSN: 2349-9028
Volume 11, Issue 1, 2024
January—April
DOI (Journal): 10.37591/JoMTRA

http://computers.stmjournals.com/index.php?journal=JoMTRA&page=index

Research JoMTRA

# Data Compression for Backbone Network

Atharva Digamber Katurde[1,*], Bhakti Bharat Shinde[2], Soham Vijay Kolapkar[3], Riya Girish Kshirsagar[4], Jitendra Musale[5], Anil Lohar[6]

**Abstract**

*This article involves the application of data compression techniques to improve the efficiency and performance of the core infrastructure of modern digital networks. This approach focuses on reducing the size of transmitted data without compromising its quality, aiming to enhance network throughput, reduce latency, and minimize energy consumption. The study also considers practical implementation challenges and trade-offs to optimize resource utilization in backbone networks. We delve into various compression methods, including lossless and lossy compression algorithms, and evaluate their effectiveness in reducing data size without compromising the quality of transmitted information. We also investigate the potential benefits of data deduplication and data pruning in further minimizing data traffic within backbone networks. We discuss the practical challenges associated with implementing data compression in backbone networks, considering issues such as real-time data processing, security, and scalability. Efficiency is a paramount concern in modern network design, and data compression plays a pivotal role in achieving this goal. We analyze the impact of data compression on network throughput, latency, and energy consumption, providing insights into how these techniques can be leveraged to meet the stringent demands of today's digital landscape.*

**Keywords:** Compression, backbone networks, infrastructure, optimization, transmission, performance, algorithms, Data compression, backbone networks, network efficiency, data transmission, network performance, compression algorithms

## INTRODUCTION

In an era driven by data and digital connectivity, backbone networks are the lifelines of our interconnected world. They carry immense volumes of information, connecting devices, data centers, and users across the globe. However, as the demand for faster data transmission, lower latency, and reduced energy consumption escalates, optimizing these networks becomes imperative.

***Author for Correspondence**
Atharva Digamber Katurde
E-mail: katurdeathu71@gmail.com

[1-4]Student, Department of Computer Engineering, ABMSP's Anantrao Pawar College of Engineering and Research Pune, Maharashtra, India
[5,6]Assistant Professor, Department of Computer Engineering, ABMSP's Anantrao Pawar College of Engineering and Research Pune, Maharashtra, India

Received Date: February 16, 2024
Accepted Date: April 02, 2024
Published Date: April 04, 2024

**Citation:** Atharva Digamber Katurde, Bhakti Bharat Shinde, Soham Vijay Kolapkar, Riya Girish Kshirsagar, Jitendra Musale, Anil Lohar. Data Compression for Backbone Network. Journal of Multimedia Technology & Recent Advancements. 2024; 11(1): 30–40p.

Data compression, a well-established technique, offers a solution to these challenges. By reducing the size of data without sacrificing quality, it can enhance network efficiency, decrease bandwidth usage, and promote energy conservation. This paper explores the application of data compression in backbone networks, evaluating its potential to shape the future of digital infrastructure.

One of the key challenges in optimizing backbone networks is the efficient utilization of network resources, including bandwidth and energy. As data volumes surge, network operators are faced with the need to increase capacity, which in turn necessitates substantial investments in infrastructure. Furthermore, the environmental impact of energy-hungry network components cannot be understated, making energy efficiency a critical concern.

The history of data compression reveals its vital role in the evolution of information technology and data transmission. From Morse code in the 1830s to modern 5G and internet of things (IoT) networks, data compression has been pivotal in optimizing data for efficient transmission and storage. Key milestones include the development of Huffman coding, early data transmission techniques like run-length encoding, and the advent of lossy compression for multimedia.

In the future, data compression will play a crucial role in backbone networks to address the escalating demands of data-intensive applications, 5G technology, and the IoT. By reducing data size, data compression will optimize network resources, increase efficiency, reduce latency, and enhance energy conservation, ensuring the seamless functioning of the digital ecosystem.

**RESEARCH**

There are various data compression algorithms which are currently used. The data compression is done based on two aspects: size of data and quality of data. Both these aspects are inversely proportionate to each other. We studied various data compression algorithms and selected the most efficient which could bring a balance in both of those aspects. The algorithms that we studied are the following:

- *Lempel-Ziv Algorithms (e.g., LZ77, LZ78):* These are widely used and form the basis for many other compression algorithms. They are effective for a broad range of data types and provide good compression ratios. The compression ratio of Lempel-Ziv algorithms depends on the nature of the input data. Suppose we have a text file that is 1000 bytes in size. After applying Lempel-Ziv algorithm, the compressed file size is 500 bytes. Then the compression ratio would be:
  Size of uncompressed data/Size of compressed data = 1000/500 = 2
  This means that the compressed file is half the size of the original file.
- *Huffman Coding:* Huffman coding is commonly used for lossless compression and is particularly efficient for compressing repetitive or redundant data. It is frequently employed in conjunction with other algorithms. The size of the input data is 11 bytes (88 bits), and the size of the compressed data is 21 bits. Therefore, the compression ratio is approximately 4.19.
- *Burrows-Wheeler Transform (BWT) with Move-to-Front (MTF) and Run-Length Encoding (RLE):* This combination is employed in algorithms like Burrows-Wheeler transform (BWT) and run-length encoded Burrows-Wheeler Transform (RLE-BWT). These algorithms perform well on certain types of data, such as text. The combination of BWT, MTF, and RLE has been shown to have an efficient compression ratio of 60.89% higher at around 0.37% compared to LZ77 2. However, the compression ratio can vary depending on the input data and the implementation of the algorithm.
- *Deflate (Used in ZIP):* Deflate combines LZ77 and Huffman coding and is widely used in the ZIP compression format. It offers a good balance between compression ratio and speed. The maximum compression ratio of the Deflate algorithm is up to 1032:1
- *Lempel-Ziv-Markov (LZMA):* LZMA, used in 7z compression, combines LZ77 with Markov models. It achieves high compression ratios but may have higher computational complexity. A hardware implementation of LZMA has shown that it is consistent over a range of input data sizes and the average compression ratio achieved is 57.135.
- *Brotli:* Developed by Google, Brotli is a relatively newer compression algorithm that aims for high compression ratios and fast decompression. It is particularly effective for web content compression. According to a blog post by Microsoft.NET, Brotli provides better compression than gzip, with JavaScript files compressed with Brotli being roughly 15% smaller, HTML files being around 20% smaller, and CSS files being around 16% smaller 23. However, the compression ratio of Brotli depends on the data set being compressed.

 

- *DEFLATE and zlib:* These are commonly used in applications like PNG image format and the HTTP protocol. They are based on LZ77 and Huffman coding. The compression ratio of the DEFLATE algorithm depends on the input data. The typical compression ratios for zlib are between 2:1 and 5:1.
- *Arithmetic Coding:* This is another entropy encoding technique that is used in various compression algorithms, such as JPEG image compression. The compression ratio of arithmetic coding can approach the optimal compression ratio of 1 - [ - 0.95 log_2(0.95) + - 0.05 log_2(0.05) ] ≈ 71.4%.

Therefore, after evaluating all these algorithms we observed that the Huffman coding algorithm is the most efficient which preserves the quality of data at the same time reducing its size significantly.

## IMPORTANCE OF TECHNOLOGY

The importance of data compression technology in backbone networks cannot be overstated. It underpins the efficiency, performance, and sustainability of these critical network infrastructures, allowing them to meet the demands of our data-driven world while minimizing costs and environmental impact.

1. *Bandwidth Efficiency:* Data compression significantly reduces the size of transmitted data, making it an indispensable tool for optimizing bandwidth usage.
2. *Latency Reduction:* Compressed data transmits faster, leading to reduced latency in network communication. This is especially crucial for real-time applications like video conferencing, online gaming, and autonomous vehicles, which rely on low-latency network connections.
3. *Energy Conservation:* Data compression contributes to energy efficiency. By transmitting smaller data payloads, the network infrastructure and connected devices can operate with reduced energy consumption, leading to cost savings and environmental benefits.
4. *Cost Savings:* Efficient data compression leads to cost savings in terms of reduced infrastructure and data storage expenses. Backbone networks can allocate resources more efficiently and cost-effectively by implementing compression techniques.
5. *Enhanced User Experience:* For end-users, data compression can lead to faster load times for web pages, smoother streaming of multimedia content, and quicker response times in applications.
6. *Security:* Data compression can also enhance data security by reducing the data's exposure to potential threats during transmission. Encrypted compressed data can be more secure in transit.
7. *Future Proofing:* As data requirements continue to evolve, data compression technology will remain relevant in adapting backbone networks to handle new and increasing data demands. It offers a future-proof solution to cope with the ever-growing digital ecosystem.

### Technology Stack Used in the Project

- Compression algorithms
- Networking protocols
- Data processing frameworks
- Middleware and application programming interfaces (APIs)
- Machine learning and artificial intelligence (AI)
- Cloud services
- Real-time data processing

## LITERATURE SURVEY

Haupt et al. [1] delved into the realm of decentralized data management for large-scale networks. They investigated several critical areas, including distributed signal processing, strategies for efficient information retrieval, the development of innovative communication protocols, data compression techniques, adaptations of information theory, and offer insights into potential future research directions within this field.

 

Das et al.’s [2] study focuses on data compression in network-on-chip (NoC) architectures for multicore systems, specifically exploring cache compression (CC) and compression in the network interface controller or NIC (NC). The research reveals that CC can reduce network latency by up to 33% with power savings of 23%, while NC leads to performance gains of up to 32% and power savings of 21%. These compression techniques also contribute to a 17% reduction in cycles per instruction (CPI), highlighting their potential to enhance the efficiency of NoC architectures.

Khade and Mini [3] discussed the challenges of data aggregation in wireless sensor networks and categorizes existing approaches into three types: tree-based, in-network, and cluster-based methods. The study suggests the potential for improving data aggregation algorithms. In this context, the authors introduce a novel approach, connected dominating set based on 1-hop Neighbors (CDS-1HN), which is utilized to establish a virtual network backbone for data aggregation.

Sadler and Martonosi [4] addressed energy efficiency challenges in sensor networks by focusing on computationally efficient lossless compression algorithms at source nodes. The paper explored the dilemma faced by sensor system designers and introduces Sensor LZW (S-LZW) and variations, demonstrating the impact of compression levels on energy savings, especially influenced by radio hardware. The study emphasizes the need for tailored compression algorithms for resource-constrained sensor nodes.

Nasif et al.'s [5] study examines networking in smart city projects, emphasizing IoT technologies as the foundation. To address the strain on network capacity from the growing number of IoT nodes, the paper proposes a compression method. Adaptive Huffman method proves the most effective, and the study suggests enhancing it with deep learning concepts to overcome IoT memory and processor limitations, ultimately improving IoT network efficiency.

Pellegrino et al.’s [6] study explored data compression in core network services for web applications, emphasizing the risk of improper implementation leading to denial of service (DoS) attacks. The research evaluated 19 implementations, identified 12 common pitfalls, and revealed undisclosed vulnerabilities that can be exploited for DoS attacks. The study underscored the importance of heightened awareness and security measures for handling compressed data in network services.

Perez-Uribe and Satizábal [7] stated the use of artificial neural networks and data compression statistics for discriminating cultured neuronal activity patterns is explored. The study applied these methods to differentiate spontaneous vs. stimulated activity patterns and responses from control and treated neuron cultures. The research highlights the potential of data compression ratios to enhance discrimination performance in the context of multi-electrode array (MEA) technology.

Alzubaidi et al. [8] explored the dominance of deep learning (DL) in machine learning across various domains, highlighting its superiority over traditional techniques. The review comprehensively covers DL fundamentals, techniques, with a focus on convolutional neural networks (CNNs). It addresses challenges, applications, and the impact of computational tools like FPGA, GPU, and CPU on DL development. The paper concludes with insights into DL's evolution and future prospects, providing a thorough overview of this transformative field.

Sangle et al. [9] in their research stated that cloud computing is often described as a network of interconnected computers and servers, serving as a foundation for various applications and services. With the widespread adoption of IoT by organizations and enterprises, there arises a significant need for rapid access to large volumes of data. However, ensuring the security of this data on the cloud is paramount to safeguard against potential threats, whether they originate from external attackers or malicious insiders. This holds particular importance in sectors such as government organizations and the IT industry.

 

## PROPOSED SYSTEM

Introducing Our Network Enhancement System: Boosting Digital Networks with Data Magic

Our proposed system is all about taking your digital network experience to the next level. We are on a mission to make networks faster, more efficient, and even more secure. Here is the lowdown, in plain and simple language:

### Our Goals

- *Speed Things Up:* We want to make networks blazing fast.
- *Use Data Wisely:* We aim to be smarter with the data we send.
- *Save Energy:* Less data means less power, so we are all for being eco-friendly.

### How We Shrink Data

You might wonder how we make data smaller without losing anything important. Well, we have got some nifty tricks up our sleeves to do just that.

### Clearing the Clutter

Imagine you are cleaning out your closet, but instead of old clothes, we are getting rid of extra, repeated data. That means your network runs even more efficiently.

### Real-Life Challenges

We know that networks can be tricky. So, we are ready to tackle real problems, like keeping your data safe and making things happen in real time.

### The Good Stuff We Bring

Zooming Networks: We will speed things up like never before. Greener Tech: With less data to send, we save energy. Lock-Tight Security: Your data stays safe and sound. Future-Proof: Our system evolves with the changing digital landscape.

### Easy Implementation

Worried about adding this to your existing network? Don't be. Our system plays nice with what you already have.

### Staying Ahead of the Game

We will not stop here. As technology keeps changing, we will keep improving our system to stay at the cutting edge. To sum it up, our system is your ticket to a faster, more efficient, and safer digital network experience. We are keeping it practical, adaptable, and ready to meet the demands of today's digital world.

## RESEARCH METHODOLOGIES

### Research Objective and Hypothesis

- *Objective:* The main aim of the project is to reduce the cost of transferring large amounts of data over the network by applying compression algorithms and reducing the size of the data being transmitted.
- *Hypothesis:* Data being transmitted should be lossless and transferred in the least amount of time saving both network cost and increasing the performance.

### Literature Review

As concerned about the existing systems, the selection of the most efficient compression algorithm has not been implemented as it is necessary. Multiple research papers helped us gain knowledge about compression algorithms and suitable data.

### Research Design

- *Research Type:* The tool will be able to provide lossless or lossy compression of data as needed to increase the quality of the data transmission by ensuring security.

 

- *Research Setting:* Areas where efficient data transmission is necessary at high speeds are where this study can come in handy.

**Data Collection**

Various types of data that can be transmitted through the network are studied and according to the type of data the most efficient compression algorithm is then selected.

**Experimental Procedure**

The data compression algorithm should be experimented with different types of data such as documents, audio, and video to see which algorithm compresses that specific data most efficiently while preserving the quality of data.

**Results and Discussion**

After analyzing the data and testing out all types of data to a specific algorithm. A single algorithm cannot be said to be the most efficient as it depends on the data.

**Conclusions and Recommendations**

We cannot conclude that any one algorithm is the most efficient. However, when algorithms used in combination with each other gave better results rather than just a single algorithm.

**Future Directions**

The implementation of artificial intelligence and machine learning can help select the best algorithm for compression based on the characteristics of data.

**Report and Presentation**

The documentation of the compression algorithms and the data that they can compress to the lowest size for efficient transmission was created to help with referencing. A presentation template was made to pitch the idea and the functioning to the users as well as customers.

**Ethical Considerations**

The guidelines mentioned in the Telecommunication Act were studied and made sure that none were being violated. The regulations in the cybersecurity laws were kept in mind during the study or experiment.

**Budget and Resources**

The budget for the implementation was estimated along with the resources required which would be the development, legal fees, etc.

## ARCHITECTURE DIAGRAM

**Data Collection and Preprocessing Unit**

This unit is responsible for gathering and preparing the data for compression. It ensures that the data is in the right format and doesn't have unnecessary elements as shown in Figure 1.

**Data Compression Unit**

This unit focuses on shrinking the data using compression techniques. It includes encoding the data and cleansing it to remove any imperfections.

**Data Gathering Unit**

Here, data is collected for the compression process. It works together with the redundancy checking unit to avoid storing duplicate information.

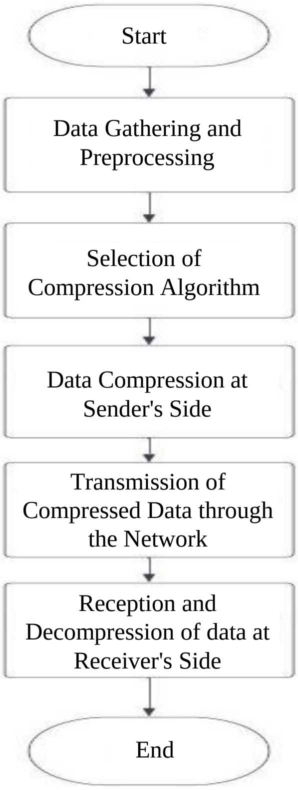


**Figure 1.** Flow diagram of the proposed system.

**Compression Algorithm Selection**

This component chooses the appropriate compression method to reduce the data size effectively as shown in Figure 2.

**Data Decompression Unit**

After data is transmitted, this unit reverses the compression process, decoding and restoring the data to its original form.

**Data Transmission Unit**

This unit manages the transfer of data and controls how it flows through the network.

**Decompression Algorithm Selection**

Similar to the compression algorithm selection, this component picks the right method to decode the data.

**Data Routing Unit**

This unit directs data to its destination within the network.

**Data Security Unit**

Ensuring data remains secure during the compression, transmission, and decompression process is the role of this component. It safeguards against unauthorized access or tampering.

**ALGORITHM**

*Step 1:* Data gathering and pre-processing
*Step 2:* Selection of compression algorithm
*Step 3:* Data compression at the sender’s side
*Step 4:* Transmission of the compressed data through the network

 

*Step 5:* Reception and decompression of the data at the receiver's side
*Step 6:* End

**Step 1: Getting Data Ready**

At the beginning, we collect the data we want to send and make sure it is neat and tidy. Think of it like organizing your stuff before packing for a trip.

**Step 2: Choosing the Right Way to Compress**

Next, we decide how to shrink the data. Imagine you have a big photo, and you can choose to make it smaller by reducing the quality a bit or keeping it as sharp as the original.

**Step 3: Compression the Data**

Now, we use the chosen method to make the data smaller. It is like using a vacuum bag to squeeze the air out of your clothes before packing them.

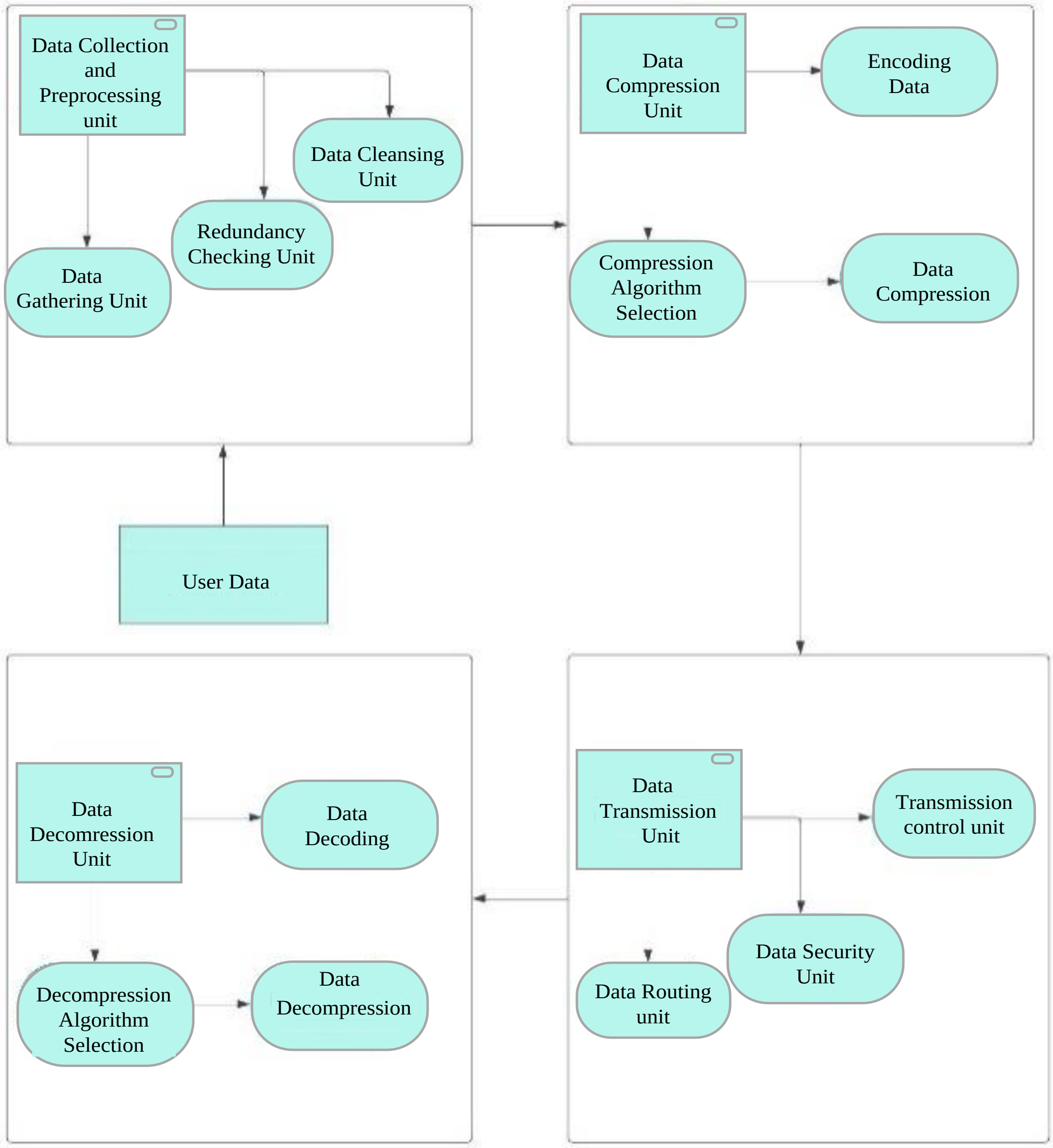


**Figure 2.** Architecture diagram of the proposed system.

 

**Step 4: Sending the Compressed Data**

The smaller data is sent through the network, like sending a compressed file through email. This saves time and space on the network.

**Step 5: Bringing the Data Back to Normal**

When the compressed data arrives at the other end, it is like opening that vacuum bag and fluffing up your clothes again. We restore the data to its original form.

**Step 6: Finished!**

Once the data is back to its original state, we are done. You can now use the data for whatever you intended, like viewing photos, watching videos, or using it in your work.

## ADVANTAGES OF THE PROPOSED MODEL

Advantages of data compression using a backbone network, like a neural network, include the following:

1. *Improved Efficiency:* These networks can extract meaningful data features, reducing data redundancy for efficient storage and transmission.
2. *Lossless or Lossy Compression:* Depending on network configuration, you can choose between lossless or lossy compression to suit your needs.
3. *Scalability:* Neural networks can adapt to handle various data sizes and complexities, making them versatile for different applications.
4. *Adaptive Compression:* Backbone networks adjust to data characteristics, optimizing compression for different data types.
5. *Compression of Various Data Types:* They can compress diverse data types such as images, text, audio, etc., making them versatile.
6. *End-to-End Compression:* Neural networks provide complete compression solutions, handling both encoding and decoding for a streamlined process.
7. *Customization:* You can fine-tune network architecture and parameters to meet specific compression requirements, offering tailored solutions.

It is essential to note that the effectiveness of data compression using a backbone network depends on factors like network architecture, training data quality, and the trade-off between compression ratio and data fidelity [10].

## CONCLUSION

In summary, leveraging neural networks as a backbone for data compression presents several advantages for users. These include improved efficiency, the flexibility to choose between lossless or lossy compression, scalability for handling diverse data sizes, adaptive compression tailored to specific data characteristics, efficient parallel processing, feature extraction capabilities, support for various data types, end-to-end compression solutions, and customizable network configurations. However, it is crucial to note that the effectiveness of this approach relies on factors such as the chosen network architecture, data quality, and the trade-off between compression ratio and data fidelity. When implemented carefully, data compression with backbone networks offers users efficient and versatile solutions for their data storage and transmission needs.

## FUTURE SCOPE

The future prospects of data compression using backbone networks, like neural networks, are quite promising. These networks have the potential for various data compression applications, including image, video, and text compression. Here are some areas of potential growth and application:

1. *Enhanced Compression Algorithms:* Backbone networks can contribute to the development of more efficient and intelligent compression algorithms. Deep learning models can adapt to the unique characteristics of different data types, leading to improved compression ratios while maintaining data quality.

 

2. *Real-Time Compression:* The increasing demand for real-time data transmission and processing can benefit from backbone networks, enabling faster and more effective compression, which is crucial for low-latency data transmission and storage.
3. *Edge Computing:* With the growth of edge computing, data compression using backbone networks becomes essential for reducing data transfer and storage requirements at the edge. This optimization can enhance the efficiency of IoT devices and resource utilization.
4. *Healthcare and Telemedicine:* In the healthcare sector, employing backbone networks for data compression can streamline the transfer of medical images and patient data, making telemedicine and remote diagnostics more accessible and efficient.
5. *Autonomous Vehicles:* Autonomous vehicles generate vast amounts of data that require quick transmission and processing. Data compression with backbone networks can efficiently manage this data, ensuring the safety and performance of autonomous systems.
6. *Multimedia Streaming:* Video and audio streaming services can benefit from advanced data compression techniques, reducing bandwidth requirements and enhancing the user experience.
7. *Data Storage:* Effective data compression can optimize data storage, making it more cost-effective and environmentally friendly, which is essential as the volume of generated data continues to rise.
8. *Security and Privacy:* Backbone networks can be harnessed to create secure and privacy-preserving compression methods, ensuring the protection of sensitive information during compression and decompression processes.
9. *Customized Compression:* Tailoring compression algorithms to specific industries and applications, such as gaming, e-commerce, or social media, can lead to more efficient data management.

In summary, the future of data compression using backbone networks holds the potential for ongoing advancements and broad applications across various domains. This is especially relevant as data continues to play a central role in our increasingly digital world, making it more intuitive, accessible, and personalized.

## REFERENCES


1. Haupt J, Bajwa WU, Rabbat M, Nowak R. Compressed sensing for networked data. IEEE Signal Process Mag. 2008; 25 (2): 92–101.
2. Das R, Mishra AK, Nicopoulos C, Park D, Narayanan V, Iyer R, Yousif MS, Das CR. Performance and power optimization through data compression in network-on-chip architectures. In: 2008 IEEE 14th International Symposium on High Performance Computer Architecture, Salt Lake City, UT, USA, February 16–20, 2008. pp. 215–225.
3. Khade G, Mini S. Identification of data aggregators in wireless sensor network. In: 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, March 22–24, 2017. pp. 1979–1983.
4. Sadler CM, Martonosi M. Data compression algorithms for energy-constrained devices in delay tolerant networks. In: SenSys '06: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, Boulder, CO, USA, October 31–November 3, 2006. pp. 265–278.
5. Nasif A, Othman ZA, Sani NS. The deep learning solutions on lossless compression methods for alleviating data load on IoT nodes in smart cities. Sensors. 2021; 21 (12): 4223.
6. Pellegrino G, Balzarotti D, Winter S, Suri N. In the compression hornet’s test: a security study of data compression in network services. In: Proceedings of the 24th USENIX Security Symposium, Washington, DC, USA, August 12–14, 2015. pp. 801–816.
7. Perez-Uribe A, Satizábal HF. Artificial neural networks and data compression statistics for the discrimination of cultured neuronal activity. In: 22nd International Conference on Artificial Neural Networks, Lausanne, Switzerland, September 11–14, 2012, Proceedings, Part I. pp. 201–208.


 

8. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L. Review of deep learning: concepts, CNN architectures, challenges, applications, and future directions. J Big Data. 2021; 8: Article 53.
9. Sangle P, Deshmukh R, Ghodake R, Yadav A, Musale J. Data security system in cloud by using fog computing and data mining. Int J Eng Computer Sci. 2016; 5 (12): 19486–19493.
10. Bomash I, Roudi Y, Nirenberg S. A virtual retina for studying population coding. PLoS One. 2013; 8 (1): e53363.