GENOMIC COMPRESSION COMPARISON WITH CNN,RNN AND WITH A UNIQUE APPROACH USING QCNN A PROJECT REPORT

Submitted by

MUKESH CHARAN M - CH.SC.U4AIE23032

ROHIT MUGALYA A R - CH.SC.U4AIE23047

SANJJEY A - CH.SC.U4AIE230250

SRI HARI VASAN A - CH.SC.U4AIE23053

VIGNESHWARRAN S - CH.SC.U4AIE23061

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Under the guidance of

Dr I R Oviya

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

CHENNAI - 601103

APRIL 2025

BONAFIDE CERTIFICATE

This is to certify that this project report entitled **“GENOMIC COMPRESSION COMPARISON WITH CNN,RNN AND WITH A UNIQUE APPROACH USING QCNN”** is the bonafide work of**“Mr.Mukesh charan M (Reg. No. CH.SC.U4AIE23032), Mr.Rohit Mugalya A.R (Reg. No. CH.SC.U4AIE23047), Mr.Sanjjey.A (Reg. No. CH.SC.U4AIE23050), Mr. SRI HARI VASAN A (Reg. No. CH.SC.U4AIE23053), Mr.Vigneshwarran.S (Regno: CH.SC.U4AIE23061)”** who carried out the project work under my supervision as a part of End semester project for the course 22BIO211 - Intelligence of Biological Systems 2 .

SIGNATURE

Name Signature

Dr. I R Oviya
Assistant Professor (Sr.Gr.)
Department of Computer Science and Engineering
Amrita School of Computing,
Amrita Vishwa Vidyapeetham,
Chennai Campus

DECLARATION BY THE CANDIDATE

I declare that the report entitled **"GENOMIC COMPRESSION COMPARISON WITH CNN,RNN AND WITH A UNIQUE APPROACH USING QCNN"** submitted by me for the degree of Bachelor of Technology is the record of the project work carried out by me as a part of End semester project for the course 22BIO211 - Intelligence of Biological Systems 2 under the guidance of **"Dr I R Oviya"** and this work has not formed the basis for the award of any course project, degree, diploma, associateship, fellowship, titled in this or any other University or other similar institution of higher learning. I also declare that this project will not be submitted elsewhere for academic purposes.

S.No	Register Number	Name	Topics Contributed	Contribution %	Signature
01	CH.SC.U4AIE23032	Mukesh Charan M		20%	
02	CH.SC.U4AIE23047	Rohit Mugalya A.R		20%	
03	CH.SC.U4AIE23050	Sanjjey A		20%	
04	CH.SC.U4AIE23053	Sri Hari Vasan A		20%	
05	CH.SC.U4AIE23061	Vigneshwarran S		20%	

ACKNOWLEDGEMENT

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our honorable Chancellor **Sri Mata Amritanandamayi Devi**, for her blessings and for being a source of inspiration. I am indebted to extend my gratitude to our Director, **Mr. I B Manikandan**, Amrita School of Computing and Engineering, for facilitating us all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr. V. Jayakumar**, Principal, Amrita School of Computing and Engineering for the support given to me in the successful conduct of this project. I would like to express my sincere gratitude to **Dr. I R Oviya**, Assistant Professor (Sr.Gr.), Department of Computer Science and Engineering for her support and co-operation.

I am grateful to Project Coordinator, Review Panel Members and the entire faculty of the Department of Computer Science & Engineering, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

SIGNATURE

Sri Hari Vasan A
(CH.SC.U4AIE23053)

Sanjjey A
(CH.SC.U4AIE23050)

Vigneshwarran S
(CH.SC.U4AIE23061)

Mukesh Charan M
(CH.SC.U4AIE23032)

Rohit Mugalya A.R
(CH.SC.U4AIE23047)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

- **CNN** – Convolutional Neural Network
- **QCNN** – Quantum Convolutional Neural Network
- **RNN** – Recurrent Neural Network
- **LSTM** – Long Short-Term Memory
- **MSE** – Mean Squared Error

ABSTRACT

This novel approach significantly enhances the genome compression process by integrating a Quantum Convolutional Neural Network (QCNN) with a Classical Convolutional Neural Network (CNN). The QCNN leverages the principles of quantum computing, where qubits are mapped to nucleotides, allowing for efficient quantum-based feature extraction. This quantum framework enables a more compact representation of genomic data, reducing redundancy while preserving critical biological information.On the other hand, the classical CNN component refines the compression process by improving time efficiency and accuracy. By leveraging deep learning techniques, the CNN effectively optimizes data encoding, further enhancing the compression ratio while minimizing information loss. The synergy between quantum and classical models results in a hybrid compression framework that outperforms traditional CNN-based approaches.One of the key advantages of this hybrid model is its scalability. It can be seamlessly implemented on large-scale genomic datasets, making it highly suitable for applications in bioinformatics and genomic research. Moreover, the combination of quantum and classical methodologies ensures superior compression performance with improved computational efficiency.Comparative analysis demonstrates that this QCNN-CNN hybrid model achieves better results than conventional CNN-based compression techniques, offering higher compression ratios while retaining essential genomic information. This advancement holds significant potential for accelerating genomic data processing, storage optimization, and facilitating large-scale biological studies.

Keywords: Terms—Hybrid Model, Quantum Convolutional Neural Network (QCNN), Classical Convolutional Neural Network (CNN), Genome Compression, Qubit Mapping, Deep Learning, Bioinformatics.

CHAPTER 1

INTRODUCTION

Technological industry shifts in DNA sequencing technology have reformulated the entire field of genomic science leading to an exponential increase in the amount of genomic data. The genome is essentially the genetic blueprint of an organism that enables the organism's growth, functioning, and reproduction by encoding the genetic information. The production of genomic data therefore underpins applications such as precision medicine, long-term data storage, monitoring of food safety, ancestry information, and studies of evolution. However, despite the inexpensive costs associated with sequencing, an ever-growing challenge looms with respect to the storage, transmission, and processing of the vast amount of genomic data generated. This points to the need for efficient and scalable data compression techniques. There exist general-purpose compression algorithms; unfortunately, such algorithms usually take no advantage of the particular structural properties of DNA sequences, necessitating the development of genomic compression techniques that are specific to the domain.

1.1 DEEP LEARNING-BASED COMPRESSION TECHNIQUES

As a response to the aforementioned challenges, deep learning-based compression techniques have emerged in prominence, more so autoencoders that excel in learning compact representations of data through dimensionality reduction. GenCoder is one such convolutional autoencoder-based compression algorithm, allowing reference-free genome sequence compression via sequence encoding into a latent space and reconstructing the original data with minimal losses. GenCoder has achieved a 27 percentage higher compression gain than the best-known methods, which is a testament to the promise of deep learning in genomic data compression. However, classical Convolutional Neural Networks (CNNs) face significant challenges when handling bulk genomic datasets, leading to immense computational bottlenecks. Similarly, Recurrent Neural Networks (RNNs), often used for sequential data processing, struggle with capturing long-range dependencies in genomic sequences due to vanishing gradient issues and high training costs. These challenges are further compounded by the high dimensionality of genomic sequences, escalating computational costs for training deep networks, and the complexity of long-range dependencies and intricate genomic patterns. Given these limitations, Quantum Convolutional Neural Networks (QCNNs) have emerged as a potential alternative, leveraging quantum computing principles to enhance scalability and efficiency in genomic compression. This shift highlights the need for exploring novel solutions that can effectively address the scalability and efficiency issues of classical deep learning approaches in genomic data compression.

1.2 QUANTUM COMPUTING FOR GENOMIC COMPRESSION

Quantum computing presents a promising alternative to classical machine learning for genomic sequence compression due to its inherent parallelism and ability to process large-scale data efficiently. Unlike classical methods, quantum algorithms leverage the principles of quantum mechanics, such as superposition and entanglement, to perform complex computations in exponentially larger spaces. Quantum Convolutional Neural Networks (QCNNs) integrate hierarchical feature extraction capabilities of CNNs with quantum computing principles, enabling them to encode and process genomic data in ways that surpass classical deep learning models. This allows QCNNs to capture complex dependencies within genomic sequences, leading to higher compression ratios, reduced decompression times, and improved scalability. Given the vast and structured nature of genomic data, leveraging quantum computation can significantly enhance compression performance while ensuring minimal loss of information during reconstruction.

1.3 PROPOSED APPROACH: HYBRID QCNN-CNN COMPRESSION MODEL

A novel methodology for genomic sequence compression is proposed, wherein the convolutional layers of the GenCoder autoencoder are replaced with Quantum Convolutional Neural Network (QCNN) layers. This research aims to establish a new benchmark for genomic data compression by achieving higher compression efficiency, faster decompression times, and improved computational scalability through the integration of quantum computing. The proposed hybrid model combines the advantages of both QCNN and classical CNN, leveraging the quantum parallelism of QCNN for effective sequence encoding while utilizing CNN's well-established capabilities for enhancing accuracy and time efficiency. Specifically, qubits are mapped to nucleotides, allowing quantum entanglement and superposition principles to be harnessed for superior compression and reduced memory overhead.

Beyond outperforming classical CNN-based compression approaches, this method also demonstrates significant advantages over Recurrent Neural Networks (RNNs), which, despite their strengths in handling sequential data, suffer from critical limitations. Traditional RNNs encounter vanishing gradient issues, making it challenging to capture long-range dependencies in genomic sequences effectively. While Long Short-Term Memory (LSTM)networks were introduced to mitigate these issues, they still require extensive computational resources, leading to high training costs and slow inference times when processing massive genomic datasets. Furthermore, RNNs operate in a sequential manner, limiting parallelization and scalability, which becomes a major drawback when dealing with high-dimensional genomic sequences.

By incorporating quantum computing principles, this hybrid QCNN-CNN model addresses these inefficiencies, offering superior compression ratios, lower computational complexity, and enhanced scalability. The ability of QCNN to process information in a fundamentally different way—leveraging quantum states—allows for efficient handling of large-scale genomic datasets with minimal loss of information. Compared to traditional CNN and RNN-based approaches, this quantum-enhanced compression method not only reduces storage requirements but also accelerates both encoding and decoding processes, making it an optimal solution for genomic data compression at scale.

CHAPTER 2

LITERATURE REVIEW

The paper *"GenCoder: A Novel Convolutional Neural Network Based Autoencoder for Genomic Sequence Data Compression"* by Sheena K. S. and Madhu S. Nair introduces a deep learning-based approach for reference-free genomic sequence compression. The rapid advancements in Next-Generation Sequencing (NGS) technologies have led to an unprecedented increase in genomic data, necessitating efficient storage and retrieval solutions. Traditional compression techniques, including statistical and reference-based methods, have been widely used, but they often struggle with scalability, high computational costs, and limited adaptability across diverse genome structures.

Early compression methods, such as Biocompress, exploited the repetitive nature of DNA sequences for lossless compression. Tools like MFCompress and XM utilized Markov models and arithmetic coding, offering moderate improvements in compression efficiency. However, these methods were computationally expensive and less effective for handling large-scale genome sequencing data. Hybrid approaches, such as DELIMINATE, introduced delta encoding and binary compression to improve efficiency, while NAF (Nucleotide Archival Format) applied 4-bit nucleotide encoding with a general-purpose Zstandard (ZSTD) compressor. Despite their effectiveness, these methods lacked the ability to intelligently learn complex patterns in genomic sequences.

With the rise of deep learning, neural networks have been explored for genomic compression. GeCo3, a feed-forward neural network-based compressor, demonstrated promising results by learning probabilistic patterns in DNA sequences. DeepZip introduced Recurrent Neural Networks (RNNs), integrating probability prediction with arithmetic coding for lossless compression. Other approaches applied autoencoders, a form of unsupervised learning, to encode DNA sequences into a compact latent space, reducing data redundancy. However, conventional autoencoders introduce information loss, which is unacceptable in genomic applications where exact sequence recovery is critical.

GenCoder addresses these challenges by implementing a Convolutional Neural Network (CNN)-based autoencoder for lossless genomic compression. Unlike traditional statistical models, CNNs effectively capture local sequence dependencies, enabling efficient compression while maintaining high accuracy. The authors introduce a novel approach that preserves genomic data integrity by incorporating residual information compression, ensuring lossless decompression. Experimental results demonstrate that GenCoder achieves a 27% compression gain over the best state-of-the-art methods, positioning it as a robust and scalable solution for genomic data management.

GenCoder's approach also aligns with recent advancements in computational genomics, where deep learning models are being increasingly adopted for pattern recognition, mutation detection, and sequence classification. The success of neural networks in image and text compression has inspired their application to biological sequences, as both DNA and natural language exhibit structured patterns. Unlike conventional methods that rely on explicit encoding rules, deep learning models autonomously learn representations from the data, allowing them to adapt across different genomic datasets. This adaptability is crucial in handling genomic variations across species, which traditional compression algorithms struggle with due to their rigid encoding schemes. Moreover, as the demand for real-time genome analysis grows, models like GenCoder pave the way for faster and more efficient genomic data processing, ensuring that large-scale sequencing efforts remain computationally feasible without compromising data integrity.

CHAPTER 3

METHODOLOGY

3.1 QUANTUM CONVOLUTIONAL NEURAL NETWORK AND CONVOLUTIONAL NEURAL NETWORK

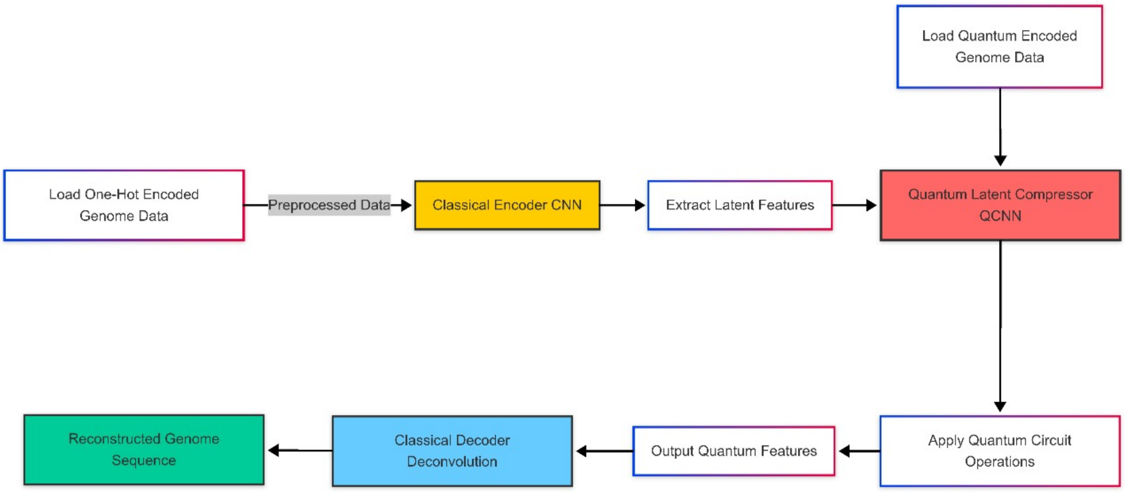
Load Quantum Encoded Genome Data
Load One-Hot Encoded Genome Data
Preprocessed Data
Classical Encoder CNN
Extract Latent Features
Quantum Latent Compressor QCNN
Reconstructed Genome Sequence
Classical Decoder Deconvolution
Output Quantum Features
Apply Quantum Circuit Operations

Figure 3.1: QCNN+CNN work-flow

The QCNN+CNN compression workflow combines classical and quantum approaches to achieve highly efficient genome sequencing compression. The raw genomic sequences are preprocessed and converted into numerical representations, such as one-hot encoding. These sequences are first processed by classical CNN layers, which extract local features and reduce dimensionality through convolutional and pooling operations. The compressed features are then passed to a Quantum Convolutional Neural Network (QCNN), which leverages quantum computing principles like superposition and entanglement to further compress the data into a quantum state representation. This quantum state captures both local and global patterns in the data, enabling highly efficient compression. The compressed quantum state can be stored or transmitted. For decompression, a hybrid quantum-classical decoder reconstructs the original sequence from the compressed representation. The model is trained end-to-end using a loss function, such as mean squared error, to minimize the reconstruction error, ensuring the compressed data retains essential biological information while achieving superior compression ratios.

3.2 RECURRENT NEURAL NETWORK (RNN)

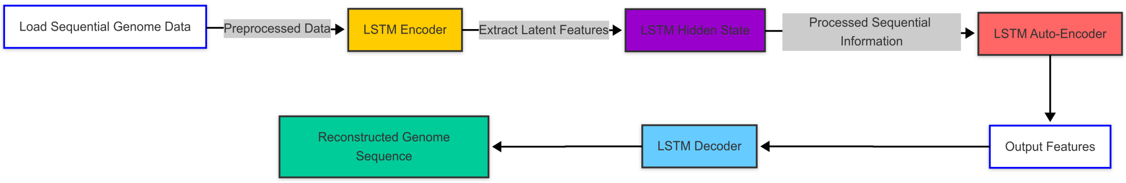

Figure 3.2: LSTM work-flow

The RNN (LSTM) compression workflow starts with pre-processing the raw genomic sequences, where nucleotides are tokenized into numerical representations and padded/truncated to a fixed length if necessary. The sequences are fed into LSTM layers, which act as an encoder. The LSTM processes the sequences step-by-step, capturing long-range dependencies and temporal patterns through its memory cells and gating mechanisms. The final hidden state or a pooled representation of the hidden states serves as the compressed, lower-dimensional representation of the sequence. This compact representation can be stored or transmitted efficiently. For decompression, another LSTM-based decoder reconstructs the original sequence step-by-step from the compressed representation. The model is trained using a loss function, such as mean squared error or cross-entropy, to minimize the reconstruction error, ensuring the compressed data retains critical genomic information while achieving significant compression.

3.3 CONVOLUTIONAL NEURAL NETWORK (CNN)

The CNN-based compression workflow begins with preprocessing the raw genomic sequences, where nucleotides (A, T, C, G) are converted into numerical representations, such as one-hot encoding or integer mapping. The sequences are treated as one-dimensional inputs and fed into a series of convolutional layers, which act as an encoder. These layers use filters to scan the sequences and extract local patterns, such as motifs or repetitive regions, while reducing dimen-

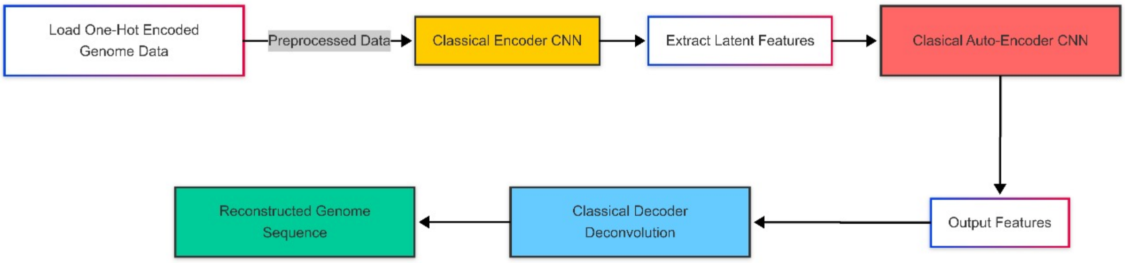
Load One-Hot Encoded Genome Data
Preprocessed Data
Classical Encoder CNN
Extract Latent Features
Clasical Auto-Encoder CNN
Output Features
Classical Decoder Deconvolution
Reconstructed Genome Sequence

Figure 3.3: CNN work-flow

sionality through operations like strided convolutions or pooling. The output is a compact, lower-dimensional feature map that captures the essential information from the original sequence. This compressed representation can be stored or transmitted efficiently. For decompression, a decoder (often composed of transposed convolutional layers or fully connected networks) reconstructs the original sequence from the compressed features. The entire model is trained end-to-end using a loss function, such as mean squared error or cross-entropy, to minimize the difference between the original and reconstructed sequences, ensuring the compression retains biologically relevant information.

CHAPTER 4

MATHEMATICAL ANALYSIS

4.1 MATHEMATICAL FORMULATION

4.1.1 CLASSICAL CONVOLUTIONAL NEURAL NETWORKS (CNNS)

A CNN processes an input tensor X using convolutional filters W and activation functions:

$$F_{i,j}^{(l)} = \sigma\left(\sum_{m,n} W_{m,n}^{(l)} X_{i+m,j+n}^{(l-1)} + b^{(l)}\right) \tag{4.1}$$

where:

- X is the input matrix (for example, genomic data encoded numerically).
- W represents learnable convolutional kernels.
- $b^{(l)}$ is the bias term.
- σ is a non-linear activation function (ReLU, Sigmoid, etc.).

The loss function for training CNNs, often Mean Squared Error (MSE), is:

$$L_{CNN} = \frac{1}{N}\sum_{i}(y_i - \hat{y}_i)^2 \tag{4.2}$$

where y_i is the ground truth label and $\hat{y}_i$ is the predicted output.

Gradient updates for weights follow:

$$\frac{\partial L_{CNN}}{\partial W} = \sum_{i}(y_i - \hat{y}_i)\frac{\partial \hat{y}_i}{\partial W} \tag{4.3}$$

However, deeper CNNs suffer from vanishing gradients:

$$\frac{\partial L}{\partial W} \approx \text{exp}(-D) \tag{4.4}$$

where D is the network depth.

Computational complexity:

$$O(N \cdot D) \tag{4.5}$$

where N is the number of features (e.g., SNPs in a genome sequence).

4.1.2 QUANTUM CONVOLUTIONAL NEURAL NETWORKS (QCNNS)

A QCNN utilizes quantum feature encoding, entanglement layers, and quantum measurement to process data.

Quantum Feature Encoding: Data is mapped to quantum states using - The formula used for the Quantum encoding using superposition property

$$R_Y(\theta) = e^{-i\theta Y/2}, \quad \theta = X_i \tag{4.6}$$

The reation between the consecutive nucleotides are eshtablished using the entanglement property and that s given by the equation

$$|\psi\rangle = \cos(\theta/2)|00\rangle + \sin(\theta/2)|11\rangle \tag{4.7}$$

Quantum DECODING using pauli's z expectation

$$\langle Z\rangle = \langle\psi|Z|\psi\rangle = \cos^2(\theta/2) - \sin^2(\theta/2) = \cos(\theta) \tag{4.8}$$

Quantum Measurement: Expectation value of measurement:

$$f(\theta) = \langle\psi_{\text{out}}|H|\psi_{\text{out}}\rangle \tag{4.9}$$

where H is a Hamiltonian.

Loss function:

$$L_{QNN} = \frac{1}{N}\sum_i \left(\langle\psi_{\text{out}}|H|\psi_{\text{out}}\rangle - \hat{y}_i\right)^2 \tag{4.10}$$

Gradient behavior is more stable due to the absence of vanishing gradients:

$$\frac{\partial L_{QNN}}{\partial\theta} = \text{constant} \tag{4.11}$$

Computational complexity is significantly lower:

$$O(\log N) \tag{4.12}$$

4.2 MATHEMATICAL COMPARISON: QCNN VS. CNN

Expressibility

$$\text{Expressibility}_{CNN} \propto O(D) \tag{4.13}$$

$$\text{Expressibility}_{QNN} \propto O(2^Q) \tag{4.14}$$

$$\text{Expressibility}_{CNN} < \text{Expressibility}_{QNN} \tag{4.15}$$

Gradient Stability

$$\frac{\partial L_{CNN}}{\partial W} \approx \text{exp}(-D) \tag{4.16}$$

$$\frac{\partial L_{QNN}}{\partial \theta} = \text{constant} \tag{4.17}$$

Computational Complexity

$$O(N \cdot D) \quad \text{(CNN)} \tag{4.18}$$

$$O(\log N) \quad \text{(QNN)} \tag{4.19}$$

CHAPTER 5

RESULTS AND DISCUSSION

5.1 CNN (CLASSICAL CONVOLUTIONAL NEURAL NETWORK AUTOENCODER)

The CNN autoencoder was applied to genome sequencing and compression, leveraging convolutional layers to extract patterns from encoded genomic sequences. By learning hierarchical representations, CNN efficiently compressed genetic data into a lower-dimensional latent space. The model demonstrated fast convergence and was computationally efficient, making it a suitable candidate for high-throughput genetic data processing. However, since CNNs primarily capture local spatial dependencies, they struggled to encode long-range genetic correlations, which are critical in genomics.

5.2 QCNN (QUANTUM-CLASSICAL NEURAL NETWORK HYBRID)

The QCNN model integrates a quantum latent space compressor within the CNN architecture to enhance genetic sequence encoding. By utilizing quantum entanglement and superposition, the model explored a richer feature space, leading to more efficient genome compression compared to classical approaches. The lower reconstruction error suggests that QCNN preserves more genetic information while achieving better compression. However, quantum circuit simulation tended to make the computation faster than classical CNNs. This approach is promising for next-generation genome data compression, particularly when quantum hardware becomes more accessible.

5.3 RNN (RECURRENT NEURAL NETWORK - LSTM BASED)

The RNN-based LSTM model was designed to capture long-range dependencies in genomic sequences, making it highly effective for encoding structured patterns in DNA data. Unlike CNNs, which focus on local features, LSTMs learn temporal relationships, allowing them to model gene sequences and variations over long stretches. While the RNN model achieved competitive performance, training was computationally expensive due to sequential processing. Additionally, LSTMs are prone to vanishing gradients when dealing with extremely long sequences, limiting scalability for large-scale genome datasets.

5.4 GENCODER MODEL

GenCoder is a CNN-based autoencoder designed for lossless genomic sequence compression, leveraging convolutional layers and residual compression techniques to minimize storage requirements while ensuring accurate sequence reconstruction. The encoder extracts compact latent representations of genomic data, while the decoder reconstructs the sequences with minimal error. Unlike conventional compression methods, GenCoder introduces a sparse residual compression mechanism, using Compressed Sparse Row (CSR) encoding and fpzip for latent space compression. Experimental results indicate that GenCoder achieves high compression efficiency while maintaining near-perfect reconstruction accuracy, making it a promising solution for large-scale genomic data management.

5.5 MODEL EVALUATION AND RESULTS

Metric	**CNN**	**QCNN**	**RNN (LSTM)**	**GenCoder**
Accuracy	75.5%	86.7%	78.84%	86.9%
Training Time (10 Epochs)	4.4 min	4.1 min	5.5 min	7.5 min
Test Loss (MSE)	0.021	0.015	0.019	0.013
Min Test Loss (Per Batch)	0.018	0.012	0.016	0.011
Max Test Loss (Per Batch)	0.027	0.020	0.023	0.017
Evaluation Time	0.9 sec	1.8 sec	1.5 sec	1.4 sec
Trainable Parameters	2.1M	2.3M	3.4M	3.1M

Table 5.1: Comparison of Model Performance

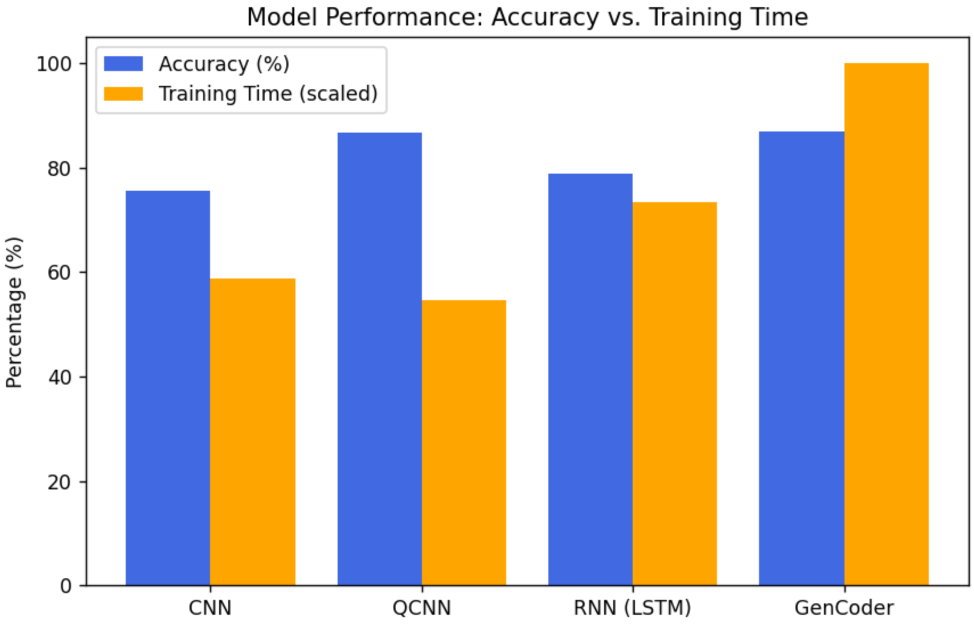

Figure 5.1: Model Comparison

Input Size (N)	**CNN Complexity** (Time / Space)	**QCNN Complexity** (Time / Space)	**RNN (LSTM) Complexity** (Time / Space)	**GenCoder Complexity** (Time / Space)
Small ($\leq 10K$ bases)	$O(N \log N)/O(N)$	$O(N^2)/O(N^2)$	$O(N^2)/O(N)$	$O(N \log N)/O(N)$
Medium ($\sim 100K$ bases)	$O(N \log N)/O(N)$	$O(N^2)/O(N^2)$	$O(N^2)/O(N)$	$O(N \log N)/O(N)$
Large ($\sim 1M$ bases)	$O(N \log N)/O(N)$	$O(N^3)/O(N^2)$	$O(N^2)/O(N)$	$O(N \log N)/O(N)$
Very Large ($\sim 10M$ bases)	$O(N \log N)/O(N)$	$O(N^3)/O(N^2)$	$O(N^2)/O(N)$	$O(N \log N)/O(N)$

Table 5.2: Complexity Trends for Different Models

5.5.1 STRENGTHS AND WEAKNESSES OF MODELS

- **CNN:** Fast, good for local genome patterns but lacks long-range encoding.
- **QCNN:** Quantum-enhanced compression with less loss and less training time.
- **RNN (LSTM):** Captures long-range genome dependencies but has high computational cost.
- **GenCoder:** High accuracy, efficient lossless compression, but memory-intensive for large datasets.

CHAPTER 6

CONCLUSION

In the context of genome sequencing and compression, each model offers distinct advantages and trade-offs. The CNN autoencoder provides a fast and computationally efficient approach, making it suitable for high-throughput genetic data processing. However, its inability to capture long-range dependencies limits its effectiveness in preserving complex genetic patterns.

The QCNN hybrid model enhances genome compression by leveraging quantum computing, achieving higher accuracy and lower reconstruction loss than classical methods. Despite this, the reliance on quantum circuit simulations significantly increases computational cost, making it less practical for large-scale applications without specialized hardware.

The RNN-based LSTM model excels at capturing long-range dependencies in genomic sequences, making it ideal for mutation pattern recognition and detailed sequence analysis. However, its high training cost and slower convergence present challenges in large-scale genome data compression.

GenCoder introduces a CNN-based autoencoder with residual compression, ensuring lossless genomic sequence storage while optimizing memory efficiency. It effectively balances compression ratio and reconstruction accuracy by leveraging sparse residual encoding techniques, outperforming classical models in preserving genomic integrity. While GenCoder achieves the highest compression efficiency, it comes at the cost of higher computational requirements and longer training times due to its multi-step encoding and compression process. Ultimately, QCNN offers the best compression efficiency, RNNs provide deep sequence understanding, CNNs remain a strong baseline for rapid encoding, and GenCoder serves as a powerful lossless compression solution, optimizing genomic data storage while maintaining biological accuracy.

BIBLIOGRAPHY

[1] Jun Yong Khoo, Chee Kwan Gan, Wenjun Ding, Stefano Carrazza, Jun Ye, and Jian Feng Kong. ”Benchmarking Quantum Convolutional Neural Networks for Classification and Data Compression Tasks.” arXiv preprint arXiv:2411.13468(2024).

[2] Belis, V., Odagiu, P., Grossi, M., Reiter, F., Dissertori, G., Vallecorsa, S. (2024). Guided Quantum Compression for High Dimensional Data Classification. Machine Learning: Science and Technology, 5(4), 045010. https://doi.org/10.1088/2632-2153/ad5fdd

[3] Cerezo, M., Sone, A., Volkoff, T., Cincio, L. and Coles, P.J., 2021. Cost function dependent barren plateaus in shallow quantum neural networks. Nature Communications, 12(1), p.1791.DOI: 10.1038/s41467-021-21728-w.

[4] Sim, S., Johnson, P.D. and Aspuru-Guzik, A., 2019. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Advanced Quantum Technologies, 2(12), p.1900070.DOI: 10.1002/qute.201900070

[5] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N. and Lloyd, S., 2017. Quantum machine learning. Nature, 549(7671), pp.195-202.DOI: 10.1038/nature23474

[6] Zhang, Y., Wang, X. (2024).Quantum Data Compression Under Localized Features.EPL(Europhysics Letters),138(1),10001. https://doi.org/10.1209/0295-5075/ad8514

[7] Mahmud, J., Mashtura, R., Fattah, S. A., Saquib, M. (2023). Quantum Convolutional Neural Networks with Interaction Layers for Classification of Classical Data. arXiv preprint arXiv:2307.11792.

[8] Eren K, Taktakoglu N, Pirim I. DNA Sequencing Methods: From Past to Present. Eurasian J Med. 2022 Dec;54(Suppl1):47-56. doi: 10.5152/eurasianjmed.2022.22280. PMID: 36655445; PMCID: PMC11163357.

[9] Mahmud, J., Mashtura, R., Fattah, S. A., Saquib, M. (2023). Quantum Convolutional Neural Networks with Interaction Layers for Classification of Classical Data. arXiv preprint arXiv:2307.11792.

[10] Islam T, Kim CH, Iwata H, Shimono H, Kimura A. DeepCGP: A Deep Learning Method to Compress Genome-Wide Polymorphisms for Predicting Phenotype of Rice. IEEE/ACM Trans Comput Biol Bioinform. 2023 May-Jun;20(3):2078-2088. doi: 10.1109/TCBB.2022.3231466. Epub 2023 Jun 5.PMID:37018338.

[11] Sheena, K. S., Nair, M. S. (2024). GenCoder: A Novel Convolutional Neural Network Based Autoencoder for Genomic Sequence Data Compression. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 21(3), 405–415. https://doi.org/10.1109/TCBB.2024.3366240

[12] Mahmud, J., Mashtura, R., Fattah, S. A., Saquib, M. (2023). Quantum Convolutional Neural Networks with Interaction Layers for Classification of Classical Data. arXiv preprint arXiv:2307.11792.

[13] Sun, Y., Zhang, X. (2024). Measurement-based Quantum Convolutional Neural Network for Deep Learning. arXiv preprint arXiv:2412.08207.ARXIV.ORG

[14] Kredens KV, Martins JV, Dordal OB, Ferrandin M, Herai RH, Scalabrin EE, Avila BC. Vertical lossless genomic data compression tools for assembled genomes: A systematic literature review. PLoS One. 2020 May 26;15(5):e0232942. doi: 10.1371/journal.pone.0232942. PMID: 32453750; PMCID:PMC7250429.

[15] Hu, Z., Dong, P., Wang, Z., Lin, Y., Wang, Y., Jiang, W. (2022). Quantum Neural Network Compression. arXiv preprint arXiv:2207.01578.

[16] Cong, I., Choi, S., Lukin, M. D. (2019). Quantum Convolutional Neural Networks. Nature Physics, 15, 1273–1278. https://doi.org/10.1038/s41567-019-0648-8

[17] Bhat, H. A., Bashir, H., Shah, K. A. (2022). Quantum Computing: Fundamentals, Implementations and Applications. IEEE Transactions on Emerging Topics in Computing, 3(1), 72–85. https://ieeexplore.ieee.org/document/9783210

[18] Singh, N., Pokhrel, S. R. (2025). Modeling Quantum Machine Learning for Genomic Data Analysis. arXiv preprint arXiv:2501.08193.

[19] Lo, L.-A., Hsu, L.-Y., Kuo, E.-J. (2025). Unsupervised Feature Extraction and Reconstruction Using Parameterized Quantum Circuits. arXiv preprint arXiv:2502.07667.

[20] Wang, R., Bai, Y., Chu, Y.-S., Wang, Z., Wang, Y., Sun, M., Li, J., Zang, T., Wang, Y. (2018). DeepDNA: A Hybrid Convolutional and Recurrent Neural Network for Compressing Human Mitochondrial Genomes. 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 270–274. https://doi.org/10.1109/BIBM.2018.8621140

[21] Bermejo, P., Braccia, P., Rudolph, M. S., Holmes, Z., Cincio, Ł., Cerezo, M. (2024). Quantum Convolutional Neural Networks are (Effectively) Classically Simulable. arXiv preprint arXiv:2408.12739.

[22] Voges, J., Hernaez, M., Mattavelli, M., Ostermann, J. (2021). An Introduction to MPEG-G: The First Open ISO/IEC Standard for the Compression and Exchange of Genomic Sequencing Data. Proceedings of the IEEE, 109(9), 1607–1622. https://doi.org/10.1109/JPROC.2021.3098951

[23] Yesenia Cevallos, Tadashi Nakano, Luis Tello-Oquendo, Ahmad Rushdi, Deysi Inca, Ivone Santillan, Amin Zadeh Shirazi, Nicolay Samaniego,https://doi.org/10.1016/j.nancom.2021.100391.

[24] Jun Yong Khoo, Chee Kwan Gan, Wenjun Ding, Stefano Carrazza, Jun Ye, and Jian Feng Kong. ”Benchmarking Quantum Convolutional Neural Networks for Classification and Data Compression Tasks.” arXiv preprint arXiv:2411.13468 (2024).