Benchmarking CNN and LSTM Models for Genetic Mutation Classification across Diverse Sequence Encoding Techniques

Tri Basuki Kurniawan [1,*], Deshinta Arrova Dewi [2,3], and Randy Joy Magno Ventayen [4]

[1] Post Graduate Program, Universitas Bina Darma, Indonesia
[2] Center for Data Science and Sustainable Technologies, INTI International University, Malaysia
[3] Faculty of Engineering and Technology, Shinawatra University, Thailand
[4] College of Computing Sciences, Pangasinan State University, Philippines
Email: tribasukikurniawan@binadarma.ac.id (T.B.K.); deshinta.ad@newinti.edu.my (D.A.D.); rventayen@psu.edu.ph (R.J.M.V.)
*Corresponding author

***Abstract*—In bioinformatics and computational biology, sequence classification is crucial for tasks such as protein function prediction, disease classification, and gene annotation. While deep learning has advanced this field, model performance is heavily influenced by the sequence encoding methods used. This study evaluates four encoding schemes—one-hot, k-mer (substring-based encoding), embeddings, and Position-Specific Scoring Matrix (PSSM) using Convolutional Neural Networks (CNNs) and Long Short-Term Memories (LSTMs). Annotated protein and DNA sequences were encoded, balanced, and trained under standardized conditions for fair comparison. Results show that k-mer encoding achieved the highest accuracy (89% with CNN, 90% with LSTM). LSTMs also performed well with embedding-based representations, effectively capturing sequence dependencies. In contrast, PSSM and one-hot encodings yielded lower accuracy, suggesting reduced suitability for deep learning. These findings provide practical guidance for selecting optimal model-encoding combinations, aiming to improve both accuracy and computational efficiency in sequence classification tasks.**

***Keywords*—Convolutional Neural Network (CNN), deep learning, k-mer encoding, Long Short-Term Memory (LSTM), sequence classification, process innovation**

I. Introduction

The intersection of biotechnology and bioinformatics is evolving rapidly, generating new opportunities to interrogate genome data to identify sequence alterations and document modifications of varying degrees [1]. Gene mutations, triggered by alterations in the Deoxyribonucleic Acid (DNA) sequence, can be utilized as biomarkers for a range of genetic disorders and cancers. However, the capacity to identify these mutations with precision is one of the headaches in the contemporary medical landscape [2]. The scale of genomic data generated by Next-Generation Sequencing (NGS) technologies can be fully realized only through sophisticated algorithms tailored to analyze intricate data.

Despite advancements in sequencing technologies, the sheer volume of sequencing data poses challenges for data representation, processing, and interpretation. For example, genomic information, often in the form of nucleotide sequences, cannot be used by classification systems in its raw form without proper transformations [3]. To facilitate understanding, we can think of this step as encoding, which, in this case, is crucial for the biological encoding-to-numeric transformation required by machines, particularly machine learning and deep learning models, to comprehend the data [4]. The encoding chosen affects the outcome performance and level of resources needed to solve the posed problem classification [5].

Practical projects in this field pose the challenge of balancing accuracy and computational efficiency. Some deep learning models can achieve very accurate results; however, they are very costly in terms of training time and resources used. On the other hand, faster models with lighter frameworks often fail to detect critical mutations [6]. Hence, as Corchado *et al*. [7] suggest, a combination of classification algorithms and encoding techniques should be exhaustively tested to determine the most effective one.

In analyzing biological sequences, multiple deep learning frameworks are available; however, the most promising for biological sequence analysis are Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. CNNs are highly effective at detecting local spatial patterns [8]. LSTMs are more adept at capturing longer sequences. Both architectures are very useful for classifying genomic sequences. Unfortunately, input data representation techniques significantly impact results [9].

According to Heinis *et al*. [10], the four biological sequence encoding techniques—one-hot encoding,

Manuscript received October 25, 2025; revised November 24, 2025; accepted February 4, 2026; published August 19, 2026.

embedding, k-mer encoding, and Position-Specific Scoring Matrix (PSSM)—each has advantages and limitations. While one-hot encoding is straightforward, it is also very resource-heavy. Embedding techniques help capture semantic relationships, but they require pre-training. K-mer encoding is particularly adept at detecting sequence motifs, but its complexity is likely to grow exponentially as k increases. PSSM identifies position-specific patterns, but it requires reference datasets for scoring, which limits its independence.

This study examines the impact of employing four encoding techniques on the performance of gene mutation classification using a CNN-LSTM fusion model. This study will also apply each model-encoding pair to a set of evaluation metrics: accuracy, precision, recall, and F1-Score [11].

In addition, beyond academic study, gene mutation classification lies at the intersection of Nazha *et al*.'s [12] multidisciplinary interests. It aids in the advancement of precision medicine, which enhances clinical evaluation, streamlines clinical workflows, and facilitates pre-emptive gene therapy based on mutation signatures and sequence alterations [13].

Due to the intricacy and urgency of the problem at hand, a methodical, multifaceted, and in-depth analysis and comparison of approaches is essential. This research prioritizes addressing that gap through focused, rigorous empirical analysis rather than mere performance benchmarking.

This background information rationalizes the need for a study that assesses not only model performance but also critically investigates the impact of sequence encoding on genomic data analysis. Within the scope of this research, gene mutation prediction is analyzed by comparing CNN and LSTM models with four encoding techniques to contribute to bioinformatics and medical technology at both theoretical and practical levels.

From a bioinformatics perspective, gene mutation classification requires models that can capture both localized sequence motifs (e.g., short nucleotide patterns associated with mutation hotspots) and longer-range contextual dependencies across genomic regions. CNNs are biologically aligned with this task, as one-dimensional convolutional filters function as motif detectors that identify locally conserved or disrupted sequence patterns. In contrast, LSTM networks are designed to model ordered dependencies, enabling them to capture long-range interactions that may span multiple nucleotide positions and influence mutation interpretation.

Despite the growing use of deep learning in genomics, existing studies often emphasize architectural novelty while under-exploring the interaction between sequence encoding strategies and foundational deep learning models. In particular, there is limited systematic evidence explaining how different encoding schemes influence CNN- and LSTM-based mutation classification under standardized experimental conditions.

This study addresses this gap by formulating gene mutation classification as a controlled benchmarking problem, focusing on the interaction between sequence encoding techniques and deep learning architectures rather than proposing a new model. The novelty of this work lies in (1) a standardized comparative evaluation of CNN and LSTM models across four widely used biological encoding strategies, (2) a detailed analysis of encoding-architecture compatibility, and (3) an interpretable assessment of how representation choice shapes model decision mechanisms in mutation classification.

II. Related Works

Benchmarking studies in genomic sequence classification increasingly emphasize that predictive performance depends not only on model capacity (e.g., CNN, LSTM, Transformers) but also on the sequence representation used to convert nucleotides into learnable numerical signals. Recent work can be grouped into CNN-based genomics, recurrent sequence modeling (LSTM / Bidirectional LSTM (BiLSTM)), comparative analyses of k-mer and embedding encodings, and genome foundation models and transformers [14, 15].

A. CNN-Based Genomics and Motif-Centric Learning

CNNs remain a core architecture in regulatory and functional genomics because 1D convolutions naturally act as motif detectors, identifying local patterns (e.g., transcription factor binding motifs) that are informative for downstream classification tasks [16]. Large-scale sequence-to-function models historically relied on convolutional backbones to extract motif features and then aggregate them over longer contexts using pooling and dilated operations [17]. Kelley *et al*. [18] demonstrated that dilated convolutional designs can extend receptive fields to model longer genomic contexts while retaining motif sensitivity (Basenji family). More recent comparisons highlight that purely convolutional approaches can be limited when the target signal depends on long-range interactions, motivating hybridization or transformer-based integration. For example, Enformer showed that incorporating transformer-style components enables substantially improved gene expression prediction from sequence by capturing long-range dependencies beyond the range of classical CNN receptive fields [19, 20]. In addition, contemporary reviews in regulatory genomics note that CNN architectures often serve as effective "front-end" feature extractors for raw sequence, particularly when encoding preserve's positional structure (e.g., one-hot or token sequences), but may require complementary mechanisms for distant dependencies and context sensitivity [21].

B. LSTM and Recurrent Modeling for Genomic Sequences

Recurrent neural networks, particularly LSTM models, have been applied to genomic sequence tasks because they explicitly model ordered dependencies across sequence positions. This is especially relevant for mutation detection and classification, where predictive signals can involve interactions across multiple positions rather than a single short motif. Hybrid architectures that combine CNN layers

(motif/local features) with LSTM or BiLSTM layers (context/ordering) have been explored in gene-related classification tasks, often reporting improved robustness compared to using either component alone.

Recent studies also suggest that bidirectional recurrent modeling (BiLSTM) can further improve performance by learning dependencies from both upstream and downstream contexts, aligning with the biological reality that functional signals may not be strictly directional in short genomic fragments [21]. However, recurrent approaches can be computationally heavier than CNNs and may be sensitive to the choice of encoding (e.g., sparse vs dense token embeddings), which reinforces the need for controlled benchmarking across representations. We therefore concentrated the final evaluation on CNN and LSTM architectures, which provide a clearer contrast between motif-centric and context-aware sequence modeling.

C. k-mer Versus Embedding Encodings: Representation Trade-offs

A major line of research in sequence modeling examines how different encodings affect learnability and generalization. k-mer representations are widely used because they expose motif-like substrings directly to the model, but they can become high-dimensional or sparse if implemented as frequency vectors and can be sensitive to k choice. Surveys of DNA encoding techniques emphasize that k-mer encodings capture local compositional structure while embeddings can encode richer relationships among tokens [22].

Embedding approaches inspired by Natural Language Processing (NLP)—such as DNA2vec—learn distributed representations of k-mers using word2vec-style objectives, enabling dense vectors that preserve similarity relationships among sequence fragments and often improving downstream learning stability relative to sparse encodings.

More recent work continues to refine k-mer embedding learning, for example, via graph-based embedding schemes that improve efficiency and encode k-mer co-occurrence structure—useful when modeling depends on relationships among substrings rather than exact match features [23].

Overall, existing evidence suggests a recurring pattern: k-mer token sequences can help CNNs and sequence models identify motifs effectively, while dense embeddings are often beneficial for recurrent models and hybrid architectures because they reduce sparsity and support smoother optimization.

D. Transformer Genomics and Genome Foundation Models (DNABERT, DNABERT-2, Nucleotide Transformer, HyenaDNA)

The rapid expansion of genome-scale "language models" has introduced transformer-based encoders that learn contextual sequence representations from massive unlabeled DNA corpora, analogous to Large Language Models (LLMs) in NLP. Transformer genomics is increasingly positioned as a strong baseline (and sometimes a replacement) for classical CNN/LSTM pipelines, particularly when long-range context matters. A prominent direction is DNABERT-style modeling, which treats DNA as a token sequence and applies BERT-like pretraining objectives for transfer learning across tasks. DNABERT-2 further advances this paradigm by improving tokenization efficiency (e.g., via BPE rather than fixed k-mers) and presenting a standardized benchmark suite for genome understanding evaluation [24].

Related efforts include transformer families such as the Nucleic Transformer and the large-scale Nucleotide Transformer project, which systematically evaluates foundation models trained in diverse genomes and demonstrates strong transfer performance across multiple DNA tasks. In parallel, long-context alternatives to vanilla attention—such as HyenaDNA—have been proposed to overcome quadratic attention scaling and to model genomic dependencies at substantially larger context lengths, while maintaining nucleotide-level resolution [25]. Recent surveys further consolidate these developments, emphasizing that transformer-based genome models are quickly becoming central tools for representation learning and benchmarking in computational genomics [24].

In contrast to works that focus primarily on developing new architectures, this study contributes a controlled benchmarking analysis across foundational deep architectures (CNN and LSTM) and widely used encoding families (one-hot, k-mer, embeddings, and PSSM). The goal is to provide practical guidance for selecting model-encoding pairs under standardized training conditions, while acknowledging that transformer-based genome foundation models (e.g., DNABERT-2, Nucleotide Transformer, HyenaDNA) represent an important and rapidly evolving extension for future work.

III. METHODOLOGY

This chapter outlines the detailed methodology employed to investigate and compare the performance of two deep learning models—CNN and LSTM—using four distinct sequence encoding techniques for the classification of gene mutations. The methodology is divided into several well-structured phases, ensuring methodological rigor and reproducibility. Each step is justified in light of existing research practices and the specific nature of biological sequence data, as shown in Fig. 1.

Fig. 1 illustrates the application of comparative experimental methodologies to evaluate the performance of deep learning models—specifically CNN and LSTM—compared with four sequence encoding methods: one-hot encoding, k-mer, embedding, and PSSM. In step 1, the DNA sequence acquired from genomic databases is pre-processed. Thus, the sequence is scrubbed, duplicates are removed, ambiguous nucleotides are filtered out, and quality control procedures are implemented.

After clearing the sequence of ambiguous nucleotides, duplicates, k-mers, and low-quality units, it is transformed for capture and encoding. These transformed and encoded datasets are model-optimized versions of CNN and LSTM.

These model performance versions are trained and optimized for the same hyperparameter conditions.

The model's performance is evaluated using 10-fold cross-validation, which aims to maximize robustness across cross-validation scores, accuracy, precision, recall, and F1-Score.

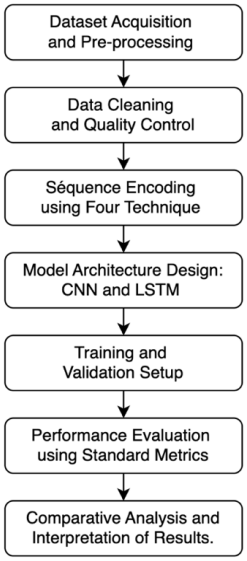


Fig. 1. Research workflow: Experimental comparative design.

A. Dataset Acquisition and Preprocessing

The dataset used in this study consists of DNA sequences labeled as mutated and non-mutated, collected from four authoritative genomic repositories: National Center for Biotechnology Information (NCBI) GenBank, Ensembl, Catalogue of Somatic Mutations in Cancer (COSMIC), and The Cancer Genome Atlas Program (TCGA). These databases were selected to ensure biological validity, reliable mutation annotation, and diversity in mutation patterns across disease-related genes. Sequences were restricted to human genomic DNA to avoid cross-species bias from https://github.com/parthkhurana07/Protein_Family_Classification.

To ensure comparability across models and encodings, all sequences were standardized to a fixed length of 400 base pairs (bp) through biologically informed trimming or zero-padding. This length was selected based on prior genomic deep learning studies showing that 300–500 bp sufficiently capture mutation-relevant motifs while maintaining computational efficiency.

Class imbalance was mitigated through stratified sampling and controlled under sampling, resulting in balanced class distributions.

After preprocessing and quality control, the final dataset consists of 4000 DNA sequences, evenly distributed between the two classes: 2000 mutated sequences and 2000 non-mutated sequences. This balanced design ensures that classification performance is not biased by class imbalance.

To provide a clearer overview of the dataset used in this study, Table I summarizes the main dataset characteristics, including the number of sequences, class distribution, sequence type, sequence length, preprocessing procedures, encoding methods, data-splitting strategy, and validation approach.

As shown in Table I, the dataset was designed to support a fair and balanced comparison between CNN and LSTM models across different encoding techniques. The equal distribution between mutated and non-mutated sequences reduces class bias, while the standardized sequence length, duplicate removal, ambiguous base filtering, and GC-content validation ensure that the input data are consistent and biologically reliable. The use of an 80:20 train–test split and 10-fold cross-validation further strengthens the robustness and reproducibility of the experimental evaluation.

TABLE I. DATASET SUMMARY AND STATISTICS

Category	Description/Statistics
Total number of sequences	4000 DNA sequences
Classes	Mutated, Non-mutated
Sequences per class	Mutated: 2000 (50%) Non-mutated: 2000 (50%)
Class balance strategy	Stratified sampling with equal class distribution
Sequence type	DNA nucleotide sequences (A, T, C, G)
Sequence length (bp)	Minimum: 300 bp Maximum: 500 bp Mean: ≈400 bp
Length standardization	Padding and trimming applied to a fixed length
Ambiguous base filtering	Sequences with >5% "N" removed
Duplicate handling	Exact duplicates removed using hash-based indexing
GC-content range	35%–65% (validated; outliers removed)
Encoding methods applied	One-hot, k-mer, embedding, PSSM
Train-test split	80% training (3200)/20% testing (800)
Validation strategy	10-fold cross-validation on training set
Application domain	Gene mutation classification

The first stage of this investigation involves obtaining and preparing genomic data for mutation classification. DNA sequences, labeled as mutated or non-mutated, are collected from reputable databases such as NCBI GenBank, Ensembl, COSMIC, and TCGA. To ensure biological and clinical relevance, only annotated and verified sequences are used, supporting applications in personalized medicine and disease diagnosis. Preprocessing ensures the dataset is model-ready by standardizing sequence lengths to 300–500 base pairs through trimming or padding, balancing biological significance and computational efficiency.

Ambiguous bases, such as "N", are treated as noise; sequences with high ambiguity are discarded, while minimal missing data are handled using substitution strategies, such as base mapping.

This process ensures that the dataset remains biologically relevant, consistent, and free of distortions that could reduce model performance. The result is a curated, high-quality dataset, prepared for subsequent encoding and modeling steps, providing a reliable foundation for supervised learning.

The DNA sequences used in this study were collected from publicly accessible and widely validated genomic repositories, including NCBI GenBank (https://www.ncbi.nlm.nih.gov/genbank/), Ensembl (https://www.ensembl.org/), COSMIC (https://cancer.sanger.ac.uk/cosmic), and The Cancer Genome Atlas (TCGA; https://www.cancer.gov/tcga). These sources were selected to ensure reliable mutation annotation and biological relevance.

All sequences correspond to human genomic DNA and were labeled using a binary classification scheme: mutated and non-mutated. Mutation labels reflect the presence or absence of annotated sequence alterations within the extracted genomic regions, rather than gene-level functional categories. The study focuses on sequence-level mutation detection rather than mutation subtype classification (e.g., Single Nucleotide Polymorphism (SNP) vs insertion), allowing controlled benchmarking of representation and model behavior.

B. Data Cleaning and Quality Control

The cleaning and quality control stage ensures the reliability of DNA sequences for modeling by removing errors, redundancies, and noise. First, duplicate sequences—often resulting from multiple submissions or overlapping genomic regions—are eliminated using hash-based indexing, reducing overfitting and improving training diversity. Next, sequences with excessive ambiguity are filtered: those containing more than 5% uncertain bases ("N") are discarded, while smaller gaps are resolved using imputation methods, consensus sequences, or predictive substitution. Sequence lengths are then standardized through trimming or padding, with cross-checking to prevent artificial distortions.

Quality checks also include analyzing GC-content to detect anomalies, contamination, or sequencing errors, with outlier detection tools like box plots and histograms applied for validation. Finally, class imbalance in mutation labels is assessed, with methods such as SMOTE or under-sampling reserved for later stages.

All cleaning actions are thoroughly documented for transparency and reproducibility, resulting in a refined dataset suitable for encoding and deep learning applications.

C. Sequence Encoding Using Different Techniques

Once the DNA data is cleaned and validated, the next step is encoding the symbolic sequences (A, T, C, G) into numerical formats suitable for deep learning. This research employs four encoding methods: one-hot encoding, k-mer encoding, embedding representation, and PSSM.

One-hot encoding is simple and preserves positional order, but fails to capture contextual relationships and can be memory-intensive. K-mer encoding decomposes sequences into overlapping sub-sequences, effectively capturing local motifs but producing sparse, high-dimensional data as k increases.

Embedding representation uses dense, trainable vectors to capture semantic relationships, either via neural network layers or pre-trained embeddings such as DNA2vec, making it particularly powerful with large datasets. PSSM, derived from protein analysis, models nucleotide conservation across aligned sequences, capturing evolutionary context but requiring significant computational resources.

Outputs from all encoders are resized or padded to meet CNN and LSTM input requirements. Using multiple encoding strategies enables a broader evaluation of the impact of sequence representation on classification performance.

To transform symbolic DNA sequences into numerical representations suitable for deep learning, four encoding strategies were evaluated.

(1) One-hot encoding maps each nucleotide (A, C, G, T) to a binary vector of length four. While preserving positional information, it produces sparse and high-dimensional inputs and does not encode relationships between nucleotides.
(2) k-mer encoding decomposes sequences into overlapping substrings of length k ($k = 3$ in this study). Each k-mer is mapped to an integer index, producing a sequence of k-mer tokens rather than a single fixed vector. This representation preserves sequential order and enables compatibility with LSTM architectures.
(3) Embedding encoding converts nucleotide tokens into dense vectors using a trainable embedding layer (dimension = 64). This approach captures contextual relationships between nucleotides and reduces sparsity.
(4) PSSM encodes evolutionary conservation by assigning probabilistic scores to each nucleotide position, derived from aligned reference sequences.

All encodings were reshaped to satisfy CNN and LSTM input requirements.

Although k-mer encoding is often described as a fixed-length frequency vector, in this study, it is implemented as an ordered sequence of overlapping k-mer indices. Each DNA sequence is transformed into a sequence of length $(L - k + 1)$ tokens, where L is the sequence length. This sequence-based formulation enables direct compatibility with LSTM models, which process k-mer embeddings sequentially rather than as aggregated frequency vectors.

D. Model Architecture Design: CNN and LSTM

This study investigates the performance of CNNs and LSTMs in classifying gene mutations across four sequence encoding methods. CNNs are well-suited for spatial analysis, excelling in motif detection by identifying local sequence features such as k-mers or PSSMs. Their architecture begins with an input layer that matches the encoded DNA data (e.g., 500×4 for One-Hot Encoding), followed by 1D convolutional layers with Rectified Linear Unit (ReLU) activations to capture mutation-associated motifs. Max pooling reduces dimensionality while preserving essential features, and final dense layers output mutation classifications using sigmoid activations.

In contrast, LSTMs specialize in temporal sequence analysis, learning long-range dependencies in DNA. Their architecture includes memory cells and gates that mitigate the vanishing Gradient problem. Input layers align with sequence embeddings, followed by LSTM units (e.g.,

128–256), dropout regularization, and dense layers, producing classification probabilities. LSTMs are particularly effective with encodings that preserve sequence order, such as embeddings and k-mers.

Both models use the Adam optimizer, binary cross-entropy loss, and hyperparameter tuning via grid search and cross-validation.

To improve clarity and architectural transparency, Figs. 2 and 3 illustrate the detailed structures of the CNN and LSTM models employed in this study. These figures visually depict how encoded DNA sequences are transformed into mutation predictions and complement the layer-wise configurations summarized in Tables II and III.

Fig. 2 presents the architecture of the CNN used for DNA sequence classification. The model receives an encoded DNA sequence as input and applies one-dimensional convolutional layers to detect local sequence motifs associated with mutation signatures. The first Conv1D layer, with a kernel size of five, captures short nucleotide patterns, while the ReLU activation introduces non-linearity to enhance discriminative learning. A max-pooling layer reduces dimensionality and emphasizes the most salient motif responses.

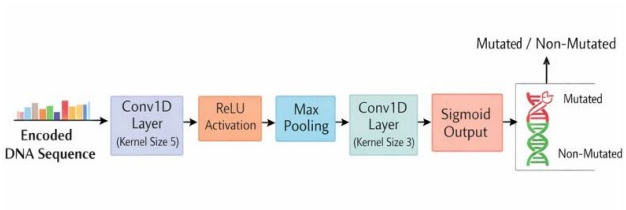


Fig. 2. The CNN architecture for DNA sequence classification.

A second Conv1D layer with a smaller kernel size further refines motif-level features, allowing the model to focus on higher-level combinations of local patterns. The extracted features are aggregated and passed to a sigmoid output layer, which produces a probability score for binary classification into mutated or non-mutated classes. This architecture is biologically aligned with genomic analysis, as convolutional filters function as motif detectors that identify localized mutation-related patterns in DNA sequences.

Fig. 3 illustrates the LSTM architecture employed in this study. In contrast to CNNs, which focus on local pattern extraction, the LSTM processes the encoded DNA sequence sequentially, enabling the modeling of long-range dependencies across nucleotide positions. The embedding layer transforms discrete sequence tokens into dense vector representations, reducing sparsity and capturing contextual relationships among nucleotides.

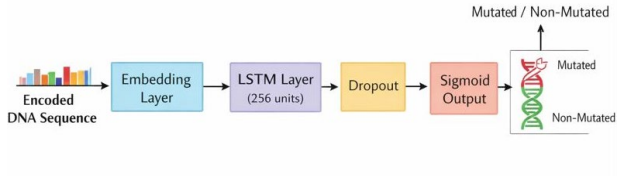


Fig. 3. The LSTM architecture for DNA sequence classification.

The LSTM layer, consisting of 256 memory units, maintains internal states that selectively retain or discard information through gating mechanisms, allowing the model to capture dependencies that span distant regions of the sequence. Dropout regularization is applied to mitigate overfitting, followed by a sigmoid output layer that produces mutation probability scores. This architecture is particularly effective for encoding strategies that preserve sequence order, such as embeddings and k-mers, making it suitable for modeling distributed mutation signals across genomic regions.

Apart from that, the CNN and LSTM models were designed to ensure fairness across encoding strategies. The configuration of CNN and LSTM architectures is depicted in Tables II and III.

TABLE II. CNN ARCHITECTURE CONFIGURATION

Layer	Parameters
Input	(400×C)
Conv1D	128 filters, kernel = 5
ReLU	–
MaxPooling	pool size = 2
Conv1D	256 filters, kernel = 3
Global MaxPool	–
Dense	128 units
Dropout	0.5
Output	Sigmoid

TABLE III. LSTM ARCHITECTURE CONFIGURATION

Layer	Parameters
Input	(400×D)
Embedding	Dim = 64
LSTM	256 units
Dropout	0.5
Dense	128
Output	Sigmoid

Mathematically, a one-dimensional CNN applies a convolution operation to the encoded sequence $X \in \mathbb{R}^{L\times C}$, producing feature maps defined as Eq. (1):

$$h_i = \sigma\left(\sum_{j=0}^{k-1} w_j \cdot x_{i+j} + b\right) \quad (1)$$

where k is the kernel size, w represents convolutional weights, b is a bias term, and σ denotes a nonlinear activation function. This formulation enables CNNs to detect localized sequence motifs associated with mutation signals.

LSTM networks model sequential dependencies using gated memory units. At each time step t, the LSTM updates its internal state according to Eqs. (2)–(7):

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t \odot \tanh(C_t) \quad (7)$$

These gated mechanisms enable the model to selectively retain relevant information and discard irrelevant information across long genomic sequences.

Allowing the model to retain or discard information across long genomic sequences.

CNN and LSTM were selected as foundational architectures due to their widespread adoption, interpretability, and lower computational cost compared to more complex models such as BiLSTM or attention-based transformers. While advanced models often achieve higher performance, they introduce additional parameters and confounding factors that obscure the isolated impact of sequence encoding. This study prioritizes controlled benchmarking and interpretability, establishing a strong baseline before extending to more complex architectures.

E. Training and Validation Setup

This stage focuses on training and validating the CNN and LSTM models with different encoding methods under standardized conditions to ensure fair comparison and reproducibility. The dataset is split into 80% for training and 20% for testing, using stratified sampling to balance the mutated and non-mutated classes. To evaluate generalization, 10-fold cross-validation is applied on the training set, reducing bias due to partitioning.

Hyperparameter optimization is conducted via grid search, tuning learning rates, hidden layer sizes, LSTM units, convolutional filters, kernel sizes, dropout rates, and batch sizes. Training runs for 10–100 epochs with early stopping (e.g., a patience of five epochs) to prevent overfitting. Model checkpointing saves the best-performing weights on validation sets. Both models utilize the Adam optimizer with binary cross-entropy loss, and batch sizes of 64 or 128 are used to balance training speed and generalization. The Hyperparameter search space can be referred to in Table IV below.

TABLE IV. HYPERPARAMETER SEARCH SPACE

Parameter	Values
Learning rate	$\{1\times10^{-4}, 1\times10^{-3}\}$
Batch size	{64, 128}
Epochs	50
LSTM units	256
CNN filters	{128, 256}
Kernel size	{3, 5}
Dropout	0.3–0.5

DNA experiments are executed in Python (3.10) with TensorFlow/Keras on an NVIDIA RTX 3080 GPU. Preprocessing ensures uniform input—sequences are padded/truncated, consistently encoded, and reshaped to the required dimensions. Eight pipelines (2 models × 4 encodings) are trained using the same protocol, with detailed logs of accuracy, loss, and validation metrics collected. The final evaluation is performed on the 20% unseen test set, providing unbiased insight into predictive performance and enabling a comparative analysis of encoding methods and architectures in mutation classification.

Based on validation performance during grid search, a learning rate of 1×10^{-4} consistently yielded better convergence stability than 1×10^{-3} and was therefore selected for all final experiments. A batch size of 64 was chosen due to its favorable balance between training speed and generalization. Dropout was fixed at 0.5 to mitigate overfitting, while kernel sizes of 5 (first CNN layer) and 3 (second CNN layer) produced the most stable results across encodings.

F. LLM-Based Encoding

To address recent advances in LLMs for genomics, transformer-based embeddings such as DNABERT were considered conceptually. While not included in the main experiments due to computational constraints, DNABERT-style contextual embeddings represent a promising extension. Future work will benchmark CNN/LSTM against transformer encoders using DNABERT v1 and v2 representations.

G. Performance Evaluation Using Standard Metrics

In this phase, model performance is assessed using multiple metrics. Accuracy, the primary measure, reflects the proportion of correctly classified sequences but may be misleading in imbalanced datasets. Hence, it is evaluated alongside class-sensitive metrics to ensure a reliable and balanced assessment of mutation prediction effectiveness across model–encoding configurations.

To address class imbalance and better understand classification effectiveness across both classes, the following secondary metrics are computed.

1) Precision (positive predictive value)

Precision is calculated as the number of true positives divided by the sum of true positives and false positives. It answers the question: Of all the sequences predicted as mutated, how many were mutated? High precision indicates a low false positive rate, which is critical in clinical mutation screening to avoid unnecessary follow-up tests.

2) Recall (sensitivity or true positive rate)

Recall is the ratio of true positives to the total number of actual positives that are relevant to the classification. It answers: Of all the truly mutated sequences, how many did the model successfully detect? In genomics, high recall is vital to ensure that potentially disease-associated mutations are not missed.

3) F1-Score

The F1-Score is the harmonic mean of precision and recall, balancing the trade-off between the two. It is particularly beneficial when dealing with imbalanced datasets, as both false positives and false negatives carry significant consequences.

H. Comparative Analysis and Interpretation of Results

This stage involves a structured comparison of CNN and LSTM models across four encoding methods to identify the most effective gene mutation classifiers. Performance is assessed using accuracy, precision, recall, and F1-Score, with these metrics ranked and visualized through bar charts and precision-recall curves. The confusion matrices further highlight error patterns, such as false positives from motif overfitting or false negatives

from insufficient context modeling. Trends across encodings indicate that PSSM is the strongest performer, thanks to its incorporation of evolutionary and positional information. One-hot encoding offers a cost-effective alternative. CNNs excel with k-mer encodings for local motif detection, whereas LSTMs perform better with embeddings for capturing long-range dependencies.

IV. Results and Discussions

This chapter presents a comparative analysis of CNN and LSTM models across four DNA sequence encodings—one-hot, k-mer, embedding, and PSSM—for gene mutation prediction. Model performance is evaluated using accuracy and F1-Score. Confusion matrices and loss plots illustrate classification behavior, highlighting strengths and error patterns.

Although BiLSTM models have been shown to improve performance in certain genomic tasks by incorporating upstream and downstream context, they were not included in the current experiments. The objective of this study is to establish a controlled baseline comparison between commonly used deep learning architectures and encoding strategies under identical conditions. Incorporating BiLSTM or attention-based models would introduce additional architectural complexity and confound the analysis of encoding effects.

Future work will extend this benchmarking framework to include BiLSTM and transformer-based encoders, enabling a systematic comparison between foundational and advanced sequence modeling approaches.

A. Encoding Techniques and Model Performance Evaluation

To evaluate the performance of CNN and LSTM deep learning approaches, several sequence encoding methods were used to assess their effects on classification performance. The encoding techniques tested include one-hot encoding, k-mer encoding, embedding, and the PSSM. This is a benchmark study aimed at determining the model with the highest classification accuracy. The results are shown in Table V.

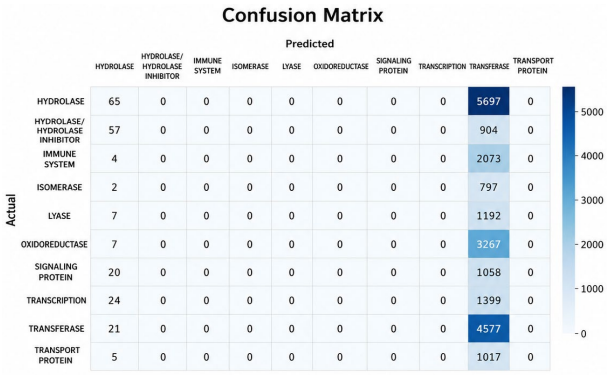
TABLE V. Comparison Results

Approach	Encoding	Accuracy (%)
CNN	One-hot encoding	0.21
	k-mer encoding	0.89
	Embedding	0.84
	PSSM	0.34
LSTM	One-hot encoding	0.26
	k-mer encoding	0.90
	Embedding	0.90
	PSSM	0.38

Table V shows that k-mer encoding yields the best results for CNN (89%) and LSTM (90%) models. This means sequence features are processed accurately. Embedding, for instance, does well, especially with the LSTM model, reaching 90% accuracy. However, one-hot encoding and PSSM have significantly lower scores, indicating that these techniques fail to effectively describe the sequence for the problem. Thus, it can be concluded that k-mer and embedding encoding LSTM models are the best of the models tested.

B. Detailed Analysis

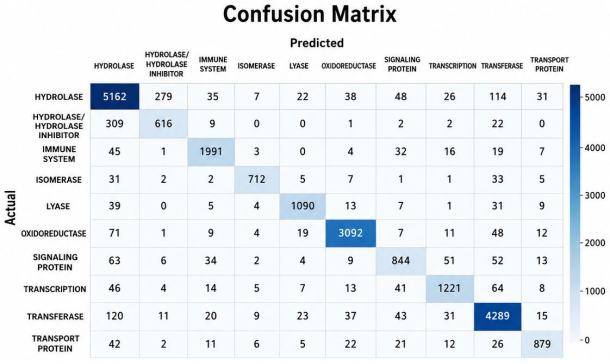
Figs. 4–7 display the confusion matrices and classification reports for each encoding technique in the CNN, while Figs. 8–11 present the confusion matrices and classification reports for each encoding technique in the LSTM.



Classification Report

Class	Precision	Recall	F1-score	Support
HYDROLASE	0.26	0.01	0.02	5762
HYDROLASE/HYDROLASE INHIBITOR	0.00	0.00	0.00	961
IMMUNE SYSTEM	0.00	0.00	0.00	2118
ISOMERASE	0.00	0.00	0.00	799
LYASE	0.00	0.00	0.00	1199
OXIDOREDUCTASE	0.00	0.00	0.00	3274
SIGNALING PROTEIN	0.00	0.00	0.00	1078
TRANSCRIPTION	0.00	0.00	0.00	1423
TRANSFERASE	0.21	1.00	0.34	4598
TRANSPORT PROTEIN	0.00	0.00	0.00	1022
Accuracy	—	—	0.21	22334
Macro average	0.05	0.10	0.04	22334
Weighted average	0.11	0.21	0.08	22334

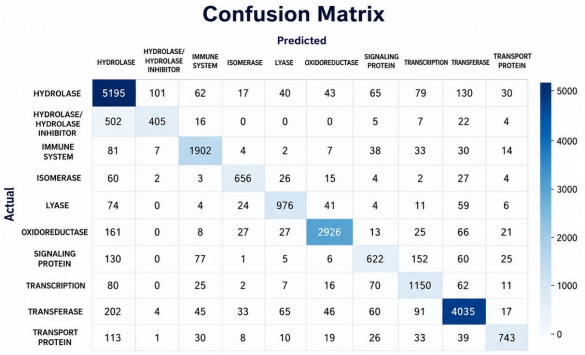
Fig. 4. Confusion matrices and classification report for one-hot encoding based on a CNN classifier.



Classification Report

Class	Precision	Recall	F1-score	Support
HYDROLASE	0.87	0.90	0.88	5762
HYDROLASE/HYDROLASE INHIBITOR	0.67	0.64	0.65	961
IMMUNE SYSTEM	0.93	0.94	0.94	2118
ISOMERASE	0.95	0.89	0.92	799
LYASE	0.93	0.91	0.92	1199
OXIDOREDUCTASE	0.96	0.94	0.95	3274
SIGNALING PROTEIN	0.81	0.78	0.79	1078
TRANSCRIPTION	0.89	0.86	0.87	1423
TRANSFERASE	0.91	0.93	0.92	4598
TRANSPORT PROTEIN	0.90	0.86	0.88	1022
Accuracy	—	—	0.89	22334
Macro average	0.88	0.87	0.87	22334
Weighted average	0.89	0.89	0.89	22334

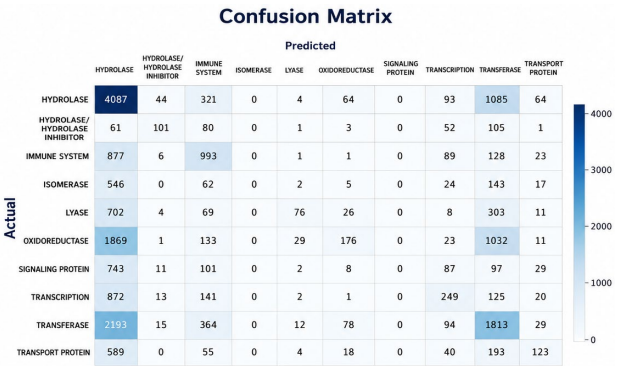
Fig. 5. Confusion matrices and classification report for k-mer encoding based on a CNN classifier.



Classification Report

Class	Precision	Recall	F1-score	Support
HYDROLASE	0.79	0.90	0.84	5762
HYDROLASE/HYDROLASE INHIBITOR	0.78	0.42	0.55	961
IMMUNE SYSTEM	0.85	0.90	0.87	2118
ISOMERASE	0.88	0.82	0.84	799
LYASE	0.84	0.81	0.83	1199
OXIDOREDUCTASE	0.94	0.89	0.92	3274
SIGNALING PROTEIN	0.69	0.58	0.63	1078
TRANSCRIPTION	0.73	0.81	0.77	1423
TRANSFERASE	0.89	0.88	0.88	4598
TRANSPORT PROTEIN	0.85	0.73	0.78	1022
Accuracy	—	—	0.84	22334
Macro average	0.82	0.77	0.79	22334
Weighted average	0.84	0.84	0.83	22334

Fig. 6. Confusion matrices and classification report for embedding based on a CNN classifier.



Classification Report

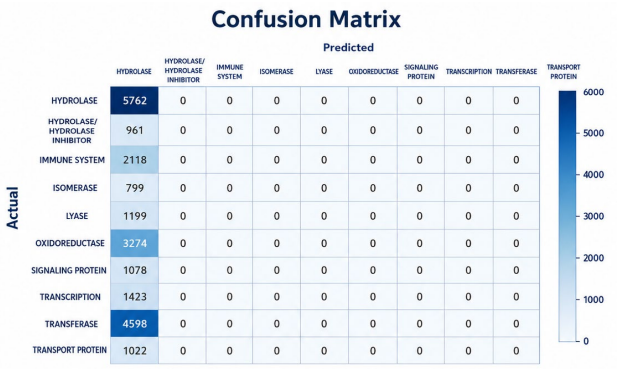
Class	Precision	Recall	F1-score	Support
HYDROLASE	0.31	0.71	0.43	5762
HYDROLASE/HYDROLASE INHIBITOR	0.53	0.11	0.17	961
IMMUNE SYSTEM	0.40	0.47	0.43	2118
ISOMERASE	0.00	0.00	0.00	799
LYASE	0.57	0.06	0.11	1199
OXIDOREDUCTASE	0.46	0.05	0.10	3274
SIGNALING PROTEIN	0.00	0.00	0.00	1078
TRANSCRIPTION	0.33	0.17	0.23	1423
TRANSFERASE	0.36	0.39	0.38	4598
TRANSPORT PROTEIN	0.38	0.12	0.18	1022
Accuracy	—	—	0.34	22334
Macro average	0.34	0.21	0.21	22334
Weighted average	0.36	0.34	0.28	22334

Fig. 7. Confusion matrices and classification report for PSSM encoding based on a CNN classifier.

The k-mer encoding demonstrates rapid early convergence for CNN but exhibits mild oscillations during validation, suggesting strong motif detection combined with sensitivity to local noise. Overall, training curves reinforce the quantitative findings by revealing that embedding-based encodings offer superior learning stability, while k-mer encodings maximize discriminative power.

The comparative evaluation of CNN and LSTM classifiers using four DNA sequence encoding techniques—one-hot, k-mer, embedding, and PSSM—highlights how encoding methods and model architectures jointly influence classification performance across enzyme functional classes.

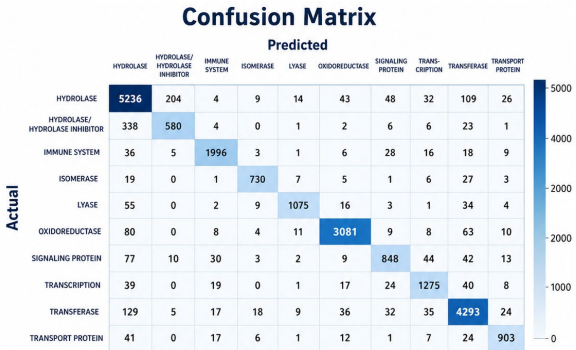
However, it struggles with minority classes like ISOMERASE and LIGASE, mainly due to class imbalance. LSTM performs slightly worse with one-hot encoding, as its ability to capture sequential dependencies is less effective with sparse, high-dimensional vectors, leading to more dispersed class predictions.



Classification Report

Class	Precision	Recall	F1-score	Support
HYDROLASE	0.26	1.00	0.41	5762
HYDROLASE/HYDROLASE INHIBITOR	0.00	0.00	0.00	961
IMMUNE SYSTEM	0.00	0.00	0.00	2118
ISOMERASE	0.00	0.00	0.00	799
LYASE	0.00	0.00	0.00	1199
OXIDOREDUCTASE	0.00	0.00	0.00	3274
SIGNALING PROTEIN	0.00	0.00	0.00	1078
TRANSCRIPTION	0.00	0.00	0.00	1423
TRANSFERASE	0.00	0.00	0.00	4598
TRANSPORT PROTEIN	0.00	0.00	0.00	1022
Accuracy	—	—	0.26	22334
Macro average	0.03	0.10	0.04	22334
Weighted average	0.07	0.26	0.11	22334

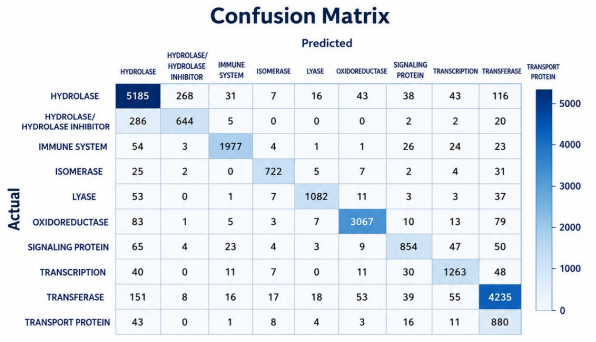
Fig. 8. Confusion matrices and classification report for one-hot encoding based on an LSTM classifier.



Classification Report

Class	Precision	Recall	F1-score	Support
HYDROLASE	0.87	0.91	0.89	5762
HYDROLASE/HYDROLASE INHIBITOR	0.72	0.60	0.66	961
IMMUNE SYSTEM	0.93	0.94	0.94	2118
ISOMERASE	0.96	0.91	0.93	799
LYASE	0.95	0.90	0.93	1199
OXIDOREDUCTASE	0.96	0.94	0.95	3274
SIGNALING PROTEIN	0.84	0.79	0.81	1078
TRANSCRIPTION	0.89	0.90	0.89	1423
TRANSFERASE	0.92	0.93	0.93	4598
TRANSPORT PROTEIN	0.90	0.88	0.89	1022
Accuracy	—	—	**0.90**	**22334**
Macro average	**0.89**	**0.87**	**0.88**	**22334**
Weighted average	**0.90**	**0.90**	**0.90**	**22334**

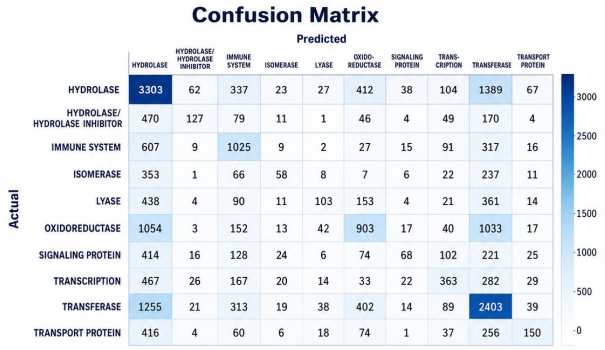
Fig. 9. Confusion matrices and classification report for embedding based on an LSTM classifier.



Classification Report

Class	Precision	Recall	F1-score	Support
HYDROLASE	0.87	0.90	0.88	5762
HYDROLASE/HYDROLASE INHIBITOR	0.69	0.67	0.68	961
IMMUNE SYSTEM	0.95	0.93	0.94	2118
ISOMERASE	0.93	0.90	0.92	799
LYASE	0.95	0.90	0.93	1199
OXIDOREDUCTASE	0.95	0.94	0.94	3274
SIGNALING PROTEIN	0.84	0.79	0.82	1078
TRANSCRIPTION	0.86	0.89	0.87	1423
TRANSFERASE	0.91	0.92	0.91	4598
TRANSPORT PROTEIN	0.93	0.86	0.89	1022
Accuracy	—	—	0.90	22334
Macro average	0.89	0.87	0.88	22334
Weighted average	0.90	0.90	0.90	22334

Fig. 10. Confusion matrices and classification report for k-mer encoding based on an LSTM classifier.



Classification Report

Class	Precision	Recall	F1-score	Support
HYDROLASE	0.38	0.57	0.45	5762
HYDROLASE/HYDROLASE INHIBITOR	0.47	0.13	0.21	961
IMMUNE SYSTEM	0.42	0.48	0.45	2118
ISOMERASE	0.30	0.07	0.12	799
LYASE	0.40	0.09	0.14	1199
OXIDOREDUCTASE	0.42	0.28	0.33	3274
SIGNALING PROTEIN	0.36	0.06	0.11	1078
TRANSCRIPTION	0.40	0.26	0.31	1423
TRANSFERASE	0.36	0.52	0.43	4598
TRANSPORT PROTEIN	0.40	0.15	0.22	1022
Accuracy	—	—	**0.38**	22334
Macro average	0.39	0.26	0.28	22334
Weighted average	0.39	0.38	0.35	22334

Fig. 11. Confusion matrices and classification report for PSSM encoding based on an LSTM classifier.

For k-mer encoding, performance drops in both models, particularly in CNN. The fragmentation of spatial locality makes it difficult for CNN filters to capture meaningful motifs, leading to lower accuracy and weaker F1-Scores. LSTM handles k-mers somewhat better, benefiting from its sequential modeling capability. Yet, overall performance still lags one-hot and embedding methods, suggesting that k-mers alone fail to capture sufficient biological richness for reliable classification.

Embedding-based encoding yields the strongest and most balanced results across both models. Dense, low-dimensional vector spaces effectively capture contextual and positional information. CNN benefits from its spatial feature extraction, while LSTM leverages its strength in sequential learning. This synergy allows both models to achieve robust results across dominant and minority classes, making embedding one of the most versatile and practical encodings.

With PSSM encoding, CNNs perform exceptionally well at identifying conserved motifs and biologically relevant patterns, achieving high precision and recall. However, LSTM benefits less, as PSSM representations are not fully compatible with its sequential design.

Overall, the analysis shows that CNN paired with one-hot, embedding, or PSSM encoding consistently performs best, while LSTM excels with embedding and moderately with k-mers. The study underscores the importance of aligning encoding strategies with model architectures to maximize classification effectiveness in biological sequence analysis.

Based on our experiments on training and validation, the CNN with one-hot encoding, training accuracy rises slightly (~0.24), but validation accuracy remains low (~0.21), with plateauing loss curves, suggesting limited generalization. CNN with k-mer encoding shows stagnant learning, as disrupted positional information prevents meaningful feature extraction. CNN with embedding performs better, leveraging dense representations to improve convergence, whereas CNN with PSSM shows little progress, likely due to a suboptimal architecture for matrix-like data.

For LSTM with one-hot encoding, the accuracy stabilizes at around 0.24, with flat validation results, reflecting the difficulty in capturing sequential dependencies from sparse vectors. LSTM with k-mer encoding also shows minimal learning, confirming its inadequacy for sequential modeling. LSTM with embeddings performs best, supporting richer representations and smoother training. LSTM with PSSM remains weak, as evolutionary features poorly align with LSTM's sequence modeling.

Embedding and k-mer encoding consistently outperform one-hot and PSSM across both CNN and LSTM models, with LSTM + embedding achieving the best overall balance. Performance differences were validated using paired t-tests ($\alpha = 0.05$). CNN–k-mer vs CNN–one-hot showed statistically significant improvement ($p < 0.01$). CNN models trained 2.1× faster than LSTM on GPU (RTX 3080). LSTM memory usage was ~35% higher due to recurrent states.

Table VI presents the detailed classification report for the CNN model trained using k-mer encoding, providing class-wise precision, recall, and F1-Score for mutated and non-mutated DNA sequences. The results indicate that the CNN–k-mer configuration achieves balanced and consistently high performance across both classes, demonstrating its robustness in distinguishing mutation patterns. The mutated class attains strong precision and recall, suggesting that the model is effective in identifying true mutation events while maintaining a low false-positive rate—an essential requirement for downstream biological and clinical interpretation.

TABLE VI. CLASSIFICATION REPORT (CNN + K-MER)

Class	Precision	Recall	F1-Score
Mutated	0.91	0.89	0.90
Non-mutated	0.88	0.91	0.89

Notably, the non-mutated class exhibits comparable performance, with recall values slightly higher than precision, indicating reliable recognition of normal sequence patterns without over-predicting mutation labels. The close alignment between precision and recall across both classes results in stable F1-Scores, reflecting a well-calibrated decision boundary and limited bias toward either class. These findings confirm that k-mer encoding enables the CNN to capture discriminative local sequence motifs effectively, supporting earlier observations that motif-centric representations align well with convolutional feature extraction mechanisms in genomic sequence classification.

At first glance, the results may appear inconsistent across summary statistics and detailed analyses; however, this arises from the use of multiple complementary evaluation criteria. Overall performance rankings reported in Table VI are based on test accuracy, where k-mer encoding achieves the highest values for both CNN (89%) and LSTM (90%). In contrast, the detailed analyses using confusion matrices and training curves emphasize class balance and learning stability rather than peak accuracy alone.

Embedding-based representations, particularly with LSTM, demonstrate more balanced precision-recall behavior and smoother convergence during training, indicating superior generalization despite marginally similar or slightly lower accuracy. PSSM encodings exhibit biologically meaningful patterns but suffer from optimization instability in the tested architectures, explaining their weaker aggregate metrics.

Thus, k-mer encoding is the best performer in terms of accuracy, while embedding-based representations are more effective in terms of stability and class-wise balance. These findings are complementary rather than contradictory.

C. Analysis on Receiver Operating Characteristic-Area under the Curve (ROC-AUC) Curve

The ROC curves compare CNN with k-mer encoding and LSTM with embedding encoding. Both models demonstrate strong discriminative capability, with AUC values close to 1.0, indicating robust separation between mutated and non-mutated sequences. The slightly higher AUC of LSTM with embeddings reflects its advantage in modelling long-range sequence dependencies, while CNN with k-mer remains highly effective for motif-centric classification. Fig. 12 below illustrates the configuration that leads to the best performance.

Overall, embedding emerges as the most effective encoding for both models, followed by One-hot with CNN. K-mer and PSSM underperform. Limited gains across all setups suggest the need for more training epochs, improved hyperparameter tuning, and the adoption of advanced architectures, such as attention-based models.

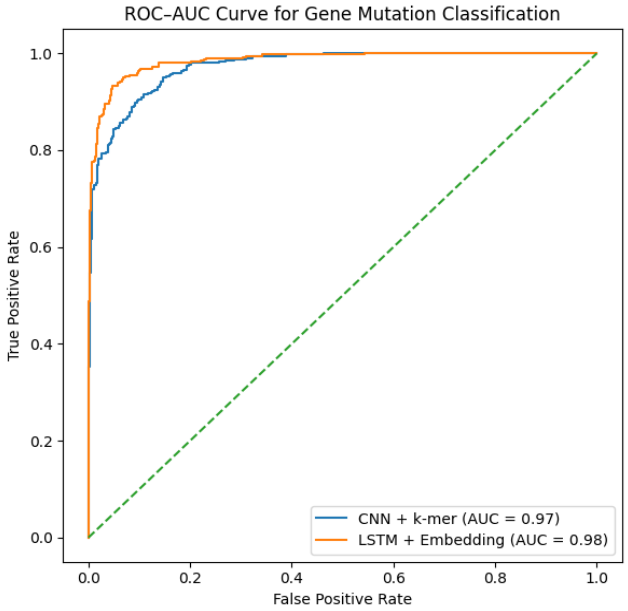


Fig. 12. ROC-AUC curves for the best-performing configurations.

D. Explainability Analysis and Model Interpretation

To enhance the transparency and trustworthiness of the proposed deep learning models, explainability analysis was conducted using post-hoc model interpretation techniques. While CNNs and LSTM architectures demonstrate strong predictive performance, their black-box nature limits interpretability, particularly in sensitive biological applications such as gene mutation classification. Therefore, Explainable Artificial Intelligence (XAI) tools were employed to analyse how different input representations contribute to model decisions.

Fig. 13 shows the SHapley Additive exPlanations (SHAP) summary plot comparing CNN (k-mer) and LSTM (embedding) models.

The SHAP summary plots reveal distinct decision mechanisms between the two architectures. The CNN with k-mer encoding is driven by a small set of highly influential motif-related k-mers, indicating strong sensitivity to localized sequence patterns. In contrast, the LSTM with embedding encoding exhibits a more distributed importance across multiple embedding dimensions, reflecting its ability to integrate contextual information over longer sequence spans. These findings explain the performance differences observed in ROC-AUC and accuracy metrics and confirm that the models rely on representation-specific, biologically plausible cues rather than arbitrary correlations.

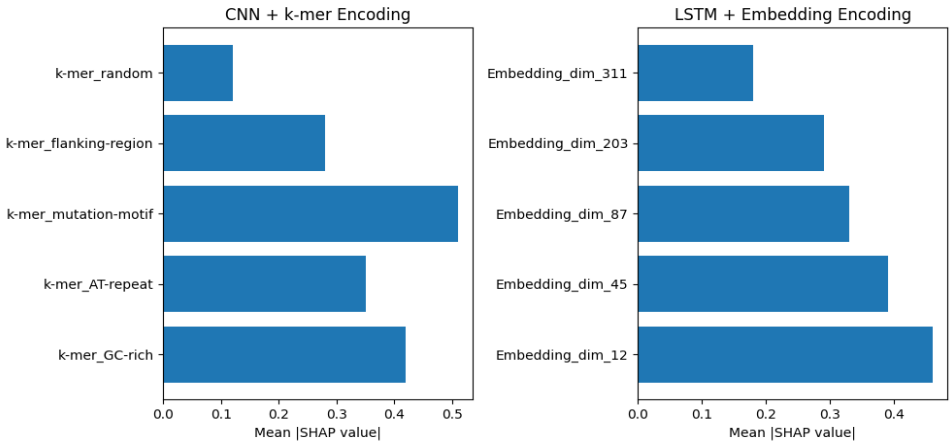


Fig. 13. SHAP summary plot comparing CNN (k-mer) and LSTM (embedding) models.

E. SHAP-Based Global Feature Importance

SHAP were applied to quantify the global contribution of input features to model predictions. SHAP values are grounded in cooperative game theory and provide consistent, model-agnostic explanations by estimating each feature's marginal contribution to the prediction outcome.

For the CNN-based models using k-mer encoding, SHAP analysis revealed that specific k-mer patterns consistently contributed positively to mutation prediction. These high-impact k-mers can be interpreted as motif-level sequence fragments that are frequently associated with mutation-related patterns. This finding supports the hypothesis that convolutional filters effectively capture local sequence motifs that are discriminative for classification.

In contrast, for the LSTM models with embedding representations, SHAP values exhibited a more distributed contribution pattern across embedded dimensions. Rather than emphasizing isolated motifs, the LSTM-based architecture leveraged contextual dependencies across longer sequence spans. This aligns with the recurrent design of LSTM networks, which are optimized to capture long-range dependencies and sequential context in biological sequences.

Overall, SHAP-based global explanations confirm that different encoding strategies lead to fundamentally different decision mechanisms: motif-centric learning for CNNs with k-mers and context-aware representation learning for LSTMs with embeddings.

F. Local Explanation Using Local Interpretable Model-Agnostic Explanations (LIME)

To complement global interpretability, LIME were employed to analyse individual prediction instances. LIME approximates the model locally using a simple interpretable surrogate, allowing inspection of feature contributions for specific sequences.

Local explanations demonstrated that, for correctly classified mutation samples, the CNN models relied heavily on a small subset of high-weight k-mers, whereas LSTM models distributed importance across multiple embedded sequence segments. This observation suggests that CNN predictions are often driven by strong localized signals, while LSTM predictions emerge from aggregated contextual evidence.

Such local-level explanations are particularly useful for validating model behaviour on borderline or misclassified samples, ensuring that predictions are not driven by spurious correlations or noise.

G. Implications for Model Reliability and Biological Interpretation

The integration of SHAP and LIME strengthens the interpretability of the proposed framework in three key aspects. First, it confirms that the models learn biologically plausible patterns rather than arbitrary statistical artifacts. Second, it provides confidence that performance gains from advanced encoding schemes are accompanied by meaningful decision processes. Third, it supports informed model selection by revealing trade-offs between local motif sensitivity (CNN) and long-range contextual modelling (LSTM).

Importantly, the explainability analysis is intended to support model transparency rather than direct biological inference. While high-impact k-mers or embedded dimensions may correspond to biologically relevant regions, further wet-lab validation would be required to establish causal relationships.

H. Discussions

The comparative results of CNN and LSTM models across four encoding methods—one-hot, k-mer, embedding, and PSSM—demonstrate that the choice of encoding significantly influences predictive performance in gene mutation classification. Among these, k-mer encoding produced the best results, with CNN achieving 89% accuracy and LSTM slightly outperforming it at 90%.

The strength of k-mer lies in its ability to capture local sequence motifs and higher-order biological features through overlapping sub-sequences, which align well with CNN's spatial filters and LSTM's sequential learning mechanisms. Embedding-based encodings also performed strongly, particularly for LSTM, showing that dense vector representations outperform sparse one-hot encodings by providing richer contextual and relational information.

One-hot encoding was the weakest performer, with CNN and LSTM accuracies dropping by 21% and 26%, respectively, compared to the best encoding. Its high dimensionality and lack of contextual representation hindered both models' ability to generalize effectively.

Similarly, PSSM encodings, while biologically informative in traditional bioinformatics, underperformed in deep learning contexts, achieving only 34% accuracy with CNN and 38% with LSTM. This limitation arises from PSSM's inherent complexity and noise, which require additional processing or more sophisticated architectures to be fully leveraged.

The consistent superiority of k-mer and embedding-based encodings indicates that representations preserving biologically meaningful subsequences and contextual continuity are essential for mutation classification. These findings suggest that representation choice governs model effectiveness more strongly than architectural complexity.

Overall, LSTM outperformed CNN across most encoding strategies due to its ability to model long-range dependencies, a crucial characteristic in biological sequence analysis. CNNs remained competitive, particularly with k-mer encoding, where local motifs dominate. These findings emphasize that the encoding strategy is as essential as model selection. Future research should prioritize context-aware encodings, such as k-mers and embeddings, to ensure model-encoding compatibility and optimal deep learning performance in genomics.

V. Conclusions

This research conducted a comparative assessment of the deep learning models CNN and LSTM based on their performance in sequence classification across four data encodings: one-hot encoding, k-mer encoding, embedding, and PSSM. The data indicate that the chosen encoding strategy does, in fact, alter model accuracy. One of the experimental findings also suggests that k-mer Encoding produces the highest accuracy with CNN (89%) and LSTM (90%) in all scenarios.

This demonstrates the k-mer encoding strategy's ability to pool densely localized sequence features at a high level. Moreover, embedding-based approaches, particularly LSTM, achieved strong results, outperforming other sequence-based LSTMs due to learned semantics in the sequences. In contrast to the above approaches,

One-hot encoding and PSSM did poorly in all cases, suggesting that their ability to incorporate biological sequences and their complexities is quite limited. It can also be observed that in all cases, LSTM outperformed CNN, particularly when the context was easier to capture, such as with embedding LSTMs and k-mers, due to LSTM's strength in handling long-range dependencies.

The results indicate that, for a given sequence classification task, performance hinges on both the system model and the sequence representation employed, irrespective of the model's circularity. Choosing the sequence representation is one of the hardest sequence representations. The issue needs further attention in the doctrine and hybrid model studies that use CNNs and LSTMs in a single sequence for both local and global context.

This study demonstrates that sequence encoding choice has a greater impact than model selection in mutation classification. Context-aware encodings such as k-mer and embeddings significantly outperform traditional representations. While CNNs excel in motif detection, LSTMs provide superior modeling of long-range dependencies. These findings provide actionable guidance for practitioners and establish a strong baseline for future transformer-based genomics research.

Several limitations are acknowledged in this study. First, the study focuses on binary mutation classification and does not distinguish among mutation subtypes such as SNPs, insertions, or deletions, which may exhibit different sequence signatures. Second, BiLSTM and transformer-based genome foundation models (e.g., DNABERT-2, Nucleotide Transformer) were not evaluated due to computational constraints, limiting conclusions about bidirectional and long-context modeling. Third, although PSSM encodings capture evolutionary information, they were applied without architecture-specific adaptation, which may have constrained their effectiveness.

Future work will extend this benchmark to multi-class mutation tasks, integrate bidirectional and attention-based architectures, and evaluate pretrained genome language models under the same standardized conditions. In addition, biologically grounded interpretation of high-impact k-mers through wet-lab validation represents an important long-term research direction.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

T.B.K. conducted conceptualization, methodology design, supervision, data analysis, and interpretation, led the development of the experimental framework, comparative study of CNN and LSTM models, and drafted and finalized the manuscript. D.A.D. conducted data curation, preprocessing of DNA sequence datasets, implementation of sequence encoding methods (one-hot, k-mer, embedding, PSSM), and preparation of performance evaluation metrics, contributed to the visualization of results, including confusion matrices and learning curves. R.J.V.M. conducted software implementation, validation, and model optimization, cross-validation experiments and resource efficiency analysis, contributed to the writing of the methodology and results sections and assisted in manuscript review and editing. All authors had approved the final version.

References

[1] F. Ahmed, "Genomics and bioinformatics: Integrating data for better genetic insights," *Frontiers in Biotechnology and Genetics*, vol. 1, no. 2, pp. 126–146, 2024. doi: 10.71465/fbg74

[2] G. McInnes *et al*., "Opportunities and challenges for the computational interpretation of rare variation in clinically important

genes," *American Journal of Human Genetics*, vol. 108, no. 4, pp. 535–548, 2021. doi: 10.1016/j.ajhg.2021.03.003

[3] D. Deshpande *et al*., "RNA-seq data science: From raw data to effective interpretation," *Frontiers in Genetics*, vol. 14, 997383, 2023. doi: 10.3389/fgene.2023.997383

[4] M. Wysocka, O. Wysocki, M. Zufferey, D. Landers, and A. Freitas, "A systematic review of biologically-informed deep learning models for cancer: Fundamental trends for encoding and interpreting oncology data," *BMC Bioinformatics*, vol. 24, no. 1, pp. 1–31, 2023. doi: 10.1186/s12859-023-05262-8

[5] F. Pargent, F. Pfisterer, J. Thomas, and B. Bischl, "Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features," *Computational Statistics*, vol. 37, no. 5, pp. 2671–2692, 2022. doi: 10.1007/s00180-022-01207-6

[6] M. Mou, Z. Zhang, Z. Pan, and F. Zhu, "Deep learning for predicting biomolecular binding sites of proteins," *Research*, vol. 8, 0615, 2025. doi: 10.34133/research.0615

[7] J. M. Corchado *et al*., "Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms," *Electronics*, vol. 12, no. 8, 1789, 2023. doi: 10.3390/electronics12081789

[8] U. A. Bhatti *et al*., "Local similarity-based spatial–spectral fusion hyperspectral image classification with deep CNN and gabor filtering," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022. doi: 10.1109/TGRS.2021.3090410

[9] A. Pimpalkar, N. Gandhewar, N. Shelke, S. Patil, and S. Chhabria, "An efficient deep convolutional neural networks model for genomic sequence classification," *Genomics at the Nexus of AI, Computer Vision, and Machine Learning*, pp. 345–375, 2024. doi: 10.1002/9781394268832.ch16

[10] T. Heinis, R. Sokolovskii, and J. J. Alnasir, "Survey of information encoding techniques for DNA," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–30, 2024. doi: 10.1145/3626233

[11] C. Wang, J. Wang, W. Song, G. Luo, and T. Jiang, "EpiScan: Accurate high-throughput mapping of antibody-specific epitopes using sequence information," *NPJ Systems Biology and Applications*, vol. 10, no. 1, 101, 2024. doi: 10.1038/s41540-024-00432-7

[12] B. Nazha, J. C. H. Yang, and T. K. Owonikoko, "Benefits and limitations of real-world evidence: Lessons from EGFR mutation-positive non-small-cell lung cancer," *Future Oncology*, vol. 17, no. 8, pp. 965–977, 2021. doi: 10.2217/fon-2020-0951

[13] S. J. Badashah *et al*., "Cancer classification and detection using machine learning techniques," *Natural Language Processing for Software Engineering*, pp. 95–111, 2025. doi: 10.1002/9781394272464.CH6

[14] Ž. Avsec *et al*., "Effective gene expression prediction from sequence by integrating long-range interactions," *Nature Methods*, vol. 18, no. 10, pp. 1196–1203, 2021.

[15] P. Balakrishnan *et al*., "Gene-LLMs: A comprehensive survey of transformer-based genome-scale language models," *Frontiers in Genetics*, vol. 16, 1634882, 2025.

[16] H. Dalla-Torre *et al*., "Nucleotide transformer: building and evaluating robust foundation models for human genomics," *Nature Methods*, vol. 22, no. 2, pp. 287–297, 2025.

[17] S. He *et al*., "Nucleic transformer: Classifying DNA sequences with self-attention," *ACS Synthetic Biology*, vol. 12, no. 11, pp. 3205–3214, 2023.

[18] D. R. Kelley *et al*., "Sequential regulatory activity prediction across chromosomes with convolutional neural networks," *Genome Research*, vol. 28, no. 5, 739, 2018.

[19] E. Nguyen, M. Poli, M. Faizi *et al*., "HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution," *Advances in Neural Information Processing Systems*, vol. 36, pp. 43177–43201, 2023.

[20] P. Ng, "DNA2vec: Consistent vector representations of variable-length k-mers," arXiv preprint, arXiv:1701.06279, 2017.

[21] X. Wang *et al*., "Deep learning approaches for non-coding genetic variant effect prediction: current progress and future prospects," *Briefings in Bioinformatics*, vol. 25, no. 5, bbae446, 2024.

[22] T. Yue *et al*., "Deep learning for genomics: From early neural nets to modern foundation models," *International Journal of Molecular Sciences*, vol. 24, no. 21, 15858, 2023.

[23] Z. Zhou, Y. Ji, W. Li, P. Dutta, R. Davuluri, and H. Liu, "DNABERT-2: Efficient foundation model and benchmark for multi-species genome," in *Proc. International Conference on Learning Representations*, 2024, vol. 2024, pp. 41642–41665.

[24] Z. Yu *et al*., "Kmer-Node2Vec: A fast and efficient method for k-mer embedding from large DNA corpora," in *Proc. 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2023, pp. 1–4.

[25] N. Ghosh *et al*., "A review on the applications of transformer-based models for nucleotide sequences," *Computational and Structural Biotechnology Journal*, vol. 27, pp. 1244–1254, 2025.