Systematic Evaluation and Bioinformatics Analysis of Machine Learning Models for PAM50-Based Breast Cancer Subtyping

University of Science and Technology Beijing
"Data Science: R Language Fundamentals" Final Project

Group Leader: Lifeng Zhang Beijing Jiaotong University
Group Member: Chenziyin Liu China University of Geosciences
Group Member: Haoming Wen China University of Geosciences
Group Member: Xingyu Liu Beijing Sport University

Date: June 21, 2025

Abstract

This study conducts a systematic evaluation and comparative analysis of various machine learning models—LightGBM, XGBoost, Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), and Support Vector Machine (SVM)—for PAM50-based breast cancer subtyping. Initially, we construct a dataset from gene expression profiles and apply five-fold cross-validation to train and evaluate each model. Next, a hybrid approach combining grid search and randomized search is employed to optimize key hyperparameters such as learning rate, number of trees, number of hidden units, and convolution filter sizes. Experimental results demonstrate that LightGBM and XGBoost achieve superior overall performance in terms of accuracy, precision, recall, and F1-score, whereas the CNN exhibits stronger robustness under class imbalance conditions. Finally, we utilize SHapley Additive exPlanations (SHAP) to visualize feature importance, revealing the biological relevance of gene expression features in distinguishing different PAM50 subtypes. The findings indicate that ensemble learning models handle high-dimensional sparse gene expression data more efficiently, deep learning models (CNN) require larger datasets and greater computational resources but perform robustly in small-sample, imbalanced scenarios, and SHAP-based interpretability analysis identifies several key genes implicated in Luminal A, Luminal B, HER2-enriched, and Basal-like subtype classification.

Keywords: breast cancer; PAM50 subtyping; machine learning; LightGBM; XGBoost; CNN; SHAP; gene expression

Group Contribution

Lifeng Zhang	Team coordination, deep learning model training, report writing
Chenziyin Liu	Data collection, data preprocessing
Haoming Wen	Data preprocessing, machine learning model training
Xingyu Liu	UI interface design

Contents

1 Introduction **1**

1.1 Breast Cancer Heterogeneity and Classification Evolution from a Molecular Subtype Perspective 1

1.1.1 Early Evolution of Breast Cancer Classification: From Pathomorphology to IHC Typing . 1

1.1.2 Discovery and Refinement of Molecular Subtypes Driven by High-Throughput Omics . . . 1

1.1.3 Establishment of the PAM50 Model and Deepening of Multi-Omics Integrated Subtyping 2

1.1.4 Fine-Grained Molecular Subtyping of Triple-Negative Breast Cancer (TNBC) and TCGA Data Resources 2

1.1.5 Significance of Molecular Subtyping for Clinical Practice and Future Challenges 3

1.2 Genomic Data Background: Application in Breast Cancer PAM50 Subtype Classification 4

1.2.1 PAM50 Gene Expression Profile: Clinical and Biological Significance 4

1.2.2 TCGA-BRCA: An Important Resource for Breast Cancer Genomics 4

1.2.3 Gene Copy Number (GCN) Analysis in Cancer 5

1.3 Machine Learning Methods for Breast Cancer Subtype Classification: Algorithmic Advances and Practical Challenges 6

1.3.1 Early Applications and Progress of Machine Learning in Breast Cancer Subtype Classification 6

1.3.2 Emergence of Innovative Machine Learning Algorithms and the Introduction of Deep Learning 6

1.3.3 Challenges and Prospects of Machine Learning in Breast Cancer Subtype Classification . 7

1.4 Multi-omics Integration and Deep Learning: New Strategies for Modeling Complex Biological Features 8

1.4.1 Necessity and Challenges of Multi-omics Integrated Analysis 8

1.4.2 Application of Deep Learning in Multi-omics Integration and Representative Models . . . 9

1.4.3 Advantages and Challenges of Multi-omics Deep Learning Methods 10

1.5 High-Dimension Low-Sample-Size (HDLSS) Problem and Data Augmentation Techniques: Generalization and Biological Signal Extraction 11

1.5.1 The High-Dimension Low-Sample-Size (HDLSS) Problem: Curse of Dimensionality and Challenges 11

1.5.2 Feature Selection and Data Dimensionality Reduction Strategies 11

1.5.3 Data Augmentation Techniques: Application of Generative Models (GANs, VAEs) 12

1.5.4 Comprehensive Strategies for Improving Generalization and Biological Signal Extraction . 13

1.6 Scope, Core Objectives, and Structure of This Report 15

2 Data Sources and Preprocessing **16**

2.1 Data Sources 16

2.2 Data Preprocessing Pipeline 17

3 Machine Learning Model Overview 19

3.1 LightGBM (Light Gradient Boosting Machine) 19

3.2 XGBoost (Extreme Gradient Boosting) 20

3.3 Multilayer Perceptron (MLP) 21

3.4 Convolutional Neural Network (CNN) 23

3.5 Support Vector Machine (SVM) 25

4 Model Implementation and Results Analysis 27

4.1 LightGBM 27

4.1.1 Implementation and Hyperparameter Tuning 27

4.1.2 Test Set Performance Evaluation 28

4.2 XGBoost 29

4.2.1 Implementation and Hyperparameter Tuning 29

4.2.2 Test Set Performance Evaluation 30

4.3 Multilayer Perceptron (MLP) 31

4.3.1 Implementation and Hyperparameter Tuning 31

4.3.2 Test Set Performance Evaluation 33

4.4 Convolutional Neural Network (CNN) 33

4.4.1 Implementation and Hyperparameter Tuning 33

4.4.2 Test Set Performance Evaluation 35

4.5 Support Vector Machine (SVM) 36

4.5.1 Implementation and Hyperparameter Tuning 36

4.5.2 Test Set Performance Evaluation 37

5 Comprehensive Evaluation, Comparison, and Strategic Recommendations 37

5.1 Cross-Model Performance Comparison for PAM50 Subtype Classification 37

5.2 Summary of Advantages and Disadvantages of Each Model in a Genomics Context 39

5.3 Recommendations for Future Research and Model Selection 40

6 Analysis of Experimental Conclusions 42

7 Conclusion 43

A Appendix: Source Code and Experimental Result 48

1 Introduction

1.1 Breast Cancer Heterogeneity and Classification Evolution from a Molecular Subtype Perspective

1.1.1 Early Evolution of Breast Cancer Classification: From Pathomorphology to IHC Typing

Breast cancer, a clinically common malignant tumor, exhibits **significant phenotypic and genetic differences**, which constitute the **core characteristic of its high heterogeneity**. It is one of the most frequently diagnosed malignant tumors in women globally, with its biological behavior showing remarkable heterogeneity across multiple levels, including histological morphology, genomic alterations, transcriptomic expression, and therapeutic response [1].

Traditional classification methods have primarily relied on immunohistochemistry (IHC) to assess the expression of estrogen receptor (ER), progesterone receptor (PR), and HER2 protein. However, IHC methods suffer from issues such as strong subjectivity and inconsistent standards, which affect the accuracy and consistency of subtyping and fail to reveal the molecular heterogeneity within subtypes [2].

Before the era of precision medicine, the early classification of breast cancer was mainly based on pathomorphological observations of tumor tissues and the expression status of key receptors (such as ER, PR, and human epidermal growth factor receptor 2, HER2). This classification method, based on IHC detection, initially divided breast cancer into major clinical subgroups like hormone receptor-positive, HER2-positive, and triple-negative. This strategy, to some extent, revealed differences in biological behavior, disease progression speed, and therapeutic choices among different breast cancer subgroups, providing a crucial basis for clinical decision-making. However, relying on a limited combination of clinicopathological features was insufficient to fully explain the finer and deeper biological diversity and underlying molecular driving mechanisms within breast cancer.

1.1.2 Discovery and Refinement of Molecular Subtypes Driven by High-Throughput Omics

The rapid development of molecular biology, especially high-throughput omics technologies, has laid the foundation for more refined molecular tumor classification. A landmark achievement was in 2000 when Perou and collaborators, using gene expression profiling, first proposed the revolutionary concept of **"intrinsic subtypes"** of breast cancer, initially classifying breast cancer into molecular subtypes such as Basal-like, HER2-enriched, Luminal A, and Luminal B [3].

Subsequently, Sørlie et al., through validation in larger independent patient cohorts and long-term clinical follow-up studies, further confirmed and refined this classification system, ultimately establishing five core intrinsic molecular subtypes: **Luminal A, Luminal B, HER2-enriched, Basal-like, and Normal-like** [4].

These molecular subtypes, defined based on whole-genome transcription profiles, show significant differences in the expression patterns of core driver genes, activation states of cell signaling pathways, clinical prognosis, and sensitivity to specific treatment regimens (such as endocrine therapy and anti-HER2 targeted therapy), profoundly revealing the inherent molecular heterogeneity of breast cancer. For example, Luminal A/B subtypes are typically ER-positive with relatively high cell differentiation, generally have a better prognosis, and are particularly responsive to endocrine therapy. In contrast, the Basal-like subtype (which highly overlaps with clinically defined triple-negative breast cancer) often exhibits stronger invasiveness and metastatic potential, with a generally poorer prognosis. The HER2-enriched subtype specifically benefits from targeted therapies against the HER2 target [5].

1.1.3 Establishment of the PAM50 Model and Deepening of Multi-Omics Integrated Subtyping

With the maturation of gene expression microarray technology and the widespread application of next-generation high-throughput sequencing (NGS), researchers have been able to analyze the molecular landscape of breast cancer in greater depth and detail. In this context, Parker et al. successfully constructed the robust **PAM50 classification model** based on the expression profile data of a carefully selected set of 50 key genes (the PAM50 gene set). This model can accurately classify tumor samples into one of the five intrinsic subtypes and has been proven effective in assisting clinical treatment decisions, such as predicting chemotherapy sensitivity and recurrence risk [5]. The advent of the PAM50 classifier and its subsequent promotion in clinical practice marked an important milestone in the transition of breast cancer molecular subtyping strategies from basic research to clinical practice.

Meanwhile, major international cancer genome collaborative research projects (such as The Cancer Genome Atlas, TCGA; and the Molecular Taxonomy of Breast Cancer International Consortium, METABRIC) have conducted the most comprehensive molecular analyses to date by integrating genomic, transcriptomic, and other multi-dimensional omics data from thousands of breast cancer patients. Their findings not only validated the existence of classic molecular subtypes but also identified more refined molecular subgroups than the traditional four- or five-category classifications. For example, Curtis and his team, after in-depth mining of genomic and transcriptomic data from over 2,000 breast tumor samples, innovatively proposed a new classification framework containing **10 Integrative Clusters (IntClusts)**, where each cluster exhibits a unique combination of genomic variation features, key signaling pathway activation patterns, and associated clinical outcomes, thus revealing a more complex and refined molecular map of breast cancer heterogeneity [6].

These cutting-edge studies collectively indicate that even within the same traditional molecular subtype, there may still be further transcriptomic differences and corresponding variations in biological behavior, suggesting that the molecular heterogeneity of breast cancer is far more complex than initially understood. In the context of precision medicine, molecular subtyping has become a key foundation for individualized treatment of breast cancer. Different molecular subtypes show significant differences in their response to targeted therapies: patients with BRCA1/2 mutations are sensitive to PARP inhibitors (like olaparib) [7], while the HER2-enriched subtype significantly benefits from anti-HER2 drug therapy (like trastuzumab) [8]. The recently proposed HER2-low subtype (IHC 1+ or IHC 2+/ISH-) is thought to have unique molecular features, such as an increased frequency of PIK3CA mutations and a trend towards EGFR amplification, suggesting potential sensitivity to PI3K/Akt pathway inhibitors [9]. These advances not only deepen the understanding of breast cancer pathogenesis but also provide new avenues for clinical precision therapy.

1.1.4 Fine-Grained Molecular Subtyping of Triple-Negative Breast Cancer (TNBC) and TCGA Data Resources

Triple-Negative Breast Cancer (TNBC) is a classic example of the high heterogeneity of breast cancer. Conventionally defined, TNBC refers to a subgroup of breast cancer that is negative for ER, PR, and HER2 expression. However, in-depth molecular studies have clearly shown that TNBC is not a homogeneous single disease entity.

Lehmann et al. were the first to meticulously subdivide TNBC based on gene expression profile features into several molecular subtypes with different biological characteristics and potential therapeutic targets, including Basal-like 1 (BL1), Basal-like 2 (BL2), Mesenchymal (M), Immunomodulatory (IM), and Luminal Androgen

Receptor (LAR). Subsequently, Burstein et al., by integrating RNA expression profiles and DNA-level genomic variation analysis, further confirmed and optimized the molecular subtyping of TNBC, proposing four more stable and clinically significant molecular subtypes: **Luminal Androgen Receptor (LAR), Mesenchymal (MES), Basal-like Immune-Suppressed (BLIS), and Basal-like Immune-Activated (BLIA)**. Research has confirmed that these four TNBC subtypes show significant differences in driver genes, signaling pathways, tumor microenvironment characteristics, and clinical responses to standard chemotherapy and emerging targeted therapies (such as PARP inhibitors and immune checkpoint inhibitors) [10].

The evolution of this classification system further highlights the complexity and diversity of the intrinsic biological behavior of breast cancer and provides an important theoretical basis for the development of individualized treatment strategies for TNBC. In terms of standardization and mechanistic research of breast cancer molecular subtyping, The Cancer Genome Atlas (TCGA) breast cancer project (TCGA-BRCA) provides a key resource. This project has integrated multi-omics data from over 1,000 breast cancer samples, including gene mutations, copy number variations, RNA expression, miRNA expression, DNA methylation, and proteomics information, accompanied by detailed clinical annotations (such as PAM50 subtyping, prognosis, and treatment regimens) [1]. Ciriello et al., based on TCGA-BRCA data, constructed a comprehensive genomic classification map of breast cancer, revealing the widespread phenomenon of ERBB2 gene amplification in the HER2-enriched subtype [11]. With its high-quality multi-modal features and standardized processing pipeline, this dataset has become a core platform for developing subtyping models, screening biomarkers, and conducting mechanistic research.

1.1.5 Significance of Molecular Subtyping for Clinical Practice and Future Challenges

The evolution of the breast cancer classification system—from initial reliance on histological morphology to the gradual introduction of IHC-based subtyping based on key protein expression, and now to the widely used molecular subtyping based on whole-genome expression profiles—has greatly deepened the scientific community's understanding of the intrinsic heterogeneity of breast cancer. Molecular subtype classification not only accurately reveals the inherent differences in core pathogenic mechanisms, natural history of the disease, and clinical prognosis among different tumor subgroups but also lays a solid foundation for achieving individualized treatment strategies based on molecular characteristics. For example, for HER2-positive breast cancer patients with HER2 gene amplification or overexpression, the successful development and application of a series of highly effective targeted drugs, such as trastuzumab and pertuzumab, have significantly improved their clinical outcomes. For ER-positive Luminal-type breast cancer, comprehensive treatment regimens centered on endocrine therapy have achieved remarkable efficacy. These brilliant research advances have collectively built the theoretical and practical framework for the fine-grained classification of breast cancer using molecular features.

However, with the explosive growth of multi-omics data and a deepening understanding of tumor complexity, new challenges have emerged: **how to effectively integrate increasingly complex and massive high-dimensional multi-omics data to further enhance the accuracy, stability, and biological and clinical value of molecular subtyping** is a key scientific question that future research urgently needs to address.

1.2 Genomic Data Background: Application in Breast Cancer PAM50 Subtype Classification

1.2.1 PAM50 Gene Expression Profile: Clinical and Biological Significance

The PAM50 gene expression profile is a molecular test involving 50 genes, widely used to classify breast cancer into five intrinsic molecular subtypes: Luminal A, Luminal B, HER2-enriched, Basal-like, and Normal-like. These subtypes not only show significant prognostic differences—for instance, Luminal A typically has the best prognosis, while HER2-enriched and Basal-like are more aggressive—but also differ in their underlying biological driver mechanisms [21]. The PAM50 test can be performed using gene expression data obtained from technologies such as microarrays, RNA sequencing (RNA-Seq), or quantitative reverse transcription-polymerase chain reaction (qRT-PCR) [21], and it has been applied in clinical practice as a prognostic tool [22].

However, PAM50 classification is not without its challenges. Studies have shown that there can be discordance between molecular subtyping based on PAM50 and surrogate subtyping methods based on immunohistochemistry (IHC) [21]. Furthermore, the preprocessing methods for gene expression data, such as modified median gene centering (MMGC) or subgroup-specific gene centering (SSGC), can influence the assignment of PAM50 subtypes [21]. This suggests that when interpreting the results of machine learning models based on PAM50, these potential sources of variation need to be considered. Interestingly, PAM50 subtypes can shift during tumor progression, for example, from primary to metastatic sites, often towards more aggressive subtypes, which has significant implications for patient prognosis [23]. Although this study primarily focuses on the classification of primary tumors, the dynamic evolution of subtypes is an important background factor for future research.

Among the five PAM50 subtypes, the existence of the ”Normal-like” subtype introduces some complexity to the construction and interpretation of machine learning models. This subtype may represent normal breast tissue mixed in with the tumor sample, or it could be a tumor entity with unique biological characteristics [21]. If the model’s training data includes the ”Normal-like” category, how it is distinguished from true normal tissue samples (if they are also included in training or as controls) and other tumor subtypes will directly affect the model’s performance metrics and the accuracy of biological interpretations. The experimental setup of this study regarding how ”Normal-like” samples are handled, or whether normal samples are included as a separate class for training, is crucial for a comprehensive understanding of model performance. Therefore, when evaluating the model, the ambiguity of the ”Normal-like” subtype and its handling should be considered for their potential impact on model performance and biological conclusions.

1.2.2 TCGA-BRCA: An Important Resource for Breast Cancer Genomics

The Cancer Genome Atlas (TCGA) project aims to systematically catalogue and discover major cancer-causing genomic alterations through large-scale genome sequencing and multi-dimensional integrated analysis, creating comprehensive genomic maps for over 30 human tumors, including breast cancer (TCGA-BRCA) [24]. TCGA data is publicly available, providing a valuable resource for researchers to understand tumorigenesis, improve diagnostic methods, and refine treatment standards [24].

The TCGA-BRCA dataset has been widely used in various machine learning-based studies. For example, some studies have used TCGA-BRCA data (analyzing 750 patients and their PAM50 subtype information) to construct gene co-expression networks based on TP53 mutation status [25]. Other studies have utilized its gene expression data for cancer type or subtype classification [26]. These applications fully demonstrate the richness of TCGA-BRCA data and its broad applicability in machine learning-driven cancer research, making it a likely source (or a similar one) for the dataset used in this study.

Although TCGA data is very comprehensive, potential batch effects are a concern due to data originating from multiple centers and possibly being collected over different time periods. If not properly addressed, batch effects could mask true biological signals, thereby interfering with the analysis results. The study by [26], when integrating TCGA and GTEx data, explicitly used the ”recount pipeline” to remove batch effects. In this study, the method of batch effect correction (if any) applied to TCGA-derived data would have a significant impact on model performance. If the data indeed comes from TCGA and the batch correction process is not detailed, this should be considered a potential variable. Different approaches to handling batch effects in different studies could also lead to discrepancies in model performance or limit the reliability of direct comparisons.

1.2.3 Gene Copy Number (GCN) Analysis in Cancer

Copy Number Variation (CNV), or Gene Copy Number (GCN), is an important class of variation in cancer genomes. GISTIC2 (Genomic Identification of Significant Targets in Cancer 2.0) is a widely used algorithm for identifying significantly amplified or deleted genomic regions across a large number of samples. The algorithm assigns a G-score to each aberration region (considering both the magnitude of the aberration and its frequency across samples) and calculates a False Discovery Rate (FDR) q-value to determine statistically significant regions [27]. GISTIC2's output includes peak regions and wide peaks for the aberration regions, with the latter being more robust for identifying the most likely target genes within a region. The `Seg.CN` column in its output files usually represents the log2 ratio of copy numbers, and values like ”1” or ”2” can indicate low-level or high-level copy number aberrations, respectively [27].

An example of GISTIC2 analysis from TCGA stomach adenocarcinoma (STAD) shows that this method can identify significant amplifications (e.g., in ERBB2, CCNE1, KRAS, MYC, CCND1, EGFR, FGFR2) or deletions in regions containing many known oncogenes and tumor suppressor genes. This highlights the biological relevance of GCN data in revealing cancer drivers.

It is important to note that the abbreviation ”GCN” can have multiple meanings in the literature. Besides gene copy number, it can also refer to Gene Correlation Networks, such as the network constructed for TCGA-BRCA patients based on TP53 mutation status in [25], which uses gene expression and survival data. Furthermore, ”GCN” can also refer to Graph Convolutional Networks/Neural Networks (GCNNs), a type of deep learning model used for processing graph-structured data, which can integrate gene expression data with information like protein-protein interaction (PPI) networks or co-expression networks for cancer subtype classification [28]. In the context of this report, when ”GCN data” is mentioned, it primarily refers to gene copy number data and its analysis.

Although this study currently focuses mainly on using gene expression data for PAM50 subtype classification, copy number variation is one of the fundamental molecular events in cancer development and often drives changes in gene expression. The results of [27] and related GISTIC2 analyses (e.g., amplifications of genes like ERBB2, MYC, CCNE1) are directly linked to the driving mechanisms of specific cancer subtypes (e.g., ERBB2 amplification is a hallmark of HER2-enriched breast cancer). The CopyClust algorithm described in [37] explicitly uses copy number data for integrated breast cancer subtyping. This suggests that integrating copy number data with gene expression data could provide a more robust and biologically meaningful classification framework, as advocated by tools like CopyClust and the broader trend of multi-omics integration [29]. A question worth exploring is whether the misclassifications produced by current expression-based models are associated with specific copy number profiles. This suggests that future research should consider the potential benefits and specific methods of integrating GCN (copy number) data with the gene expression data used by existing models.

1.3 Machine Learning Methods for Breast Cancer Subtype Classification: Algorithmic Advances and Practical Challenges

1.3.1 Early Applications and Progress of Machine Learning in Breast Cancer Subtype Classification

With the rapid advancements in bioinformatics technology and the continuous development of Machine Learning (ML) theories and algorithms, researchers are increasingly inclined to use advanced computational methods to achieve automated, high-precision discrimination of breast cancer molecular subtypes. Throughout its development, the application of relevant algorithms in breast cancer subtype classification tasks has seen significant progress and widespread use, from early clustering algorithms based on exploratory data analysis and relatively simple traditional classifiers to the introduction of more structurally complex and powerful modern machine learning models (especially deep learning models) in recent years.

The initial methods for classifying intrinsic breast cancer subtypes (such as the classic PAM50 classifier) can be understood as a simplified machine learning model based on a nearest neighbor or prototype-based (e.g., Nearest Centroid) classification strategy, using distances between samples' gene expression profiles (like Euclidean distance or correlation distance) [5]. Building on this, a series of classic supervised learning algorithms, such as **Support Vector Machines (SVM), Random Forest (RF), and k-Nearest Neighbors (kNN)**, have been widely applied to process high-dimensional gene expression data to effectively distinguish between different molecular subtypes [12].

For example, Wu et al. used large-scale breast cancer transcriptomic data from the TCGA database to systematically train and compare multiple traditional machine learning models, including SVM, kNN, Naive Bayes, and Decision Trees, for distinguishing between triple-negative and non-triple-negative breast cancer patients. The study found that the SVM model showed the highest classification accuracy in this specific task [12]. Such studies initially confirmed the feasibility and potential of traditional machine learning methods in extracting effective discriminative information from high-dimensional gene expression profiles and applying it to breast cancer subtype discrimination tasks. In recent years, researchers have widely used TCGA-BRCA data in conjunction with machine learning and deep learning methods for breast cancer subtyping. For instance, Tong et al. proposed a multi-omics fusion model based on Graph Neural Networks (GNN), which outperformed traditional methods in subtype identification [13]. These algorithms demonstrate the great potential of deep models in revealing complex biological patterns.

1.3.2 Emergence of Innovative Machine Learning Algorithms and the Introduction of Deep Learning

In recent years, innovative algorithms specifically designed for breast cancer subtype classification have continuously emerged. These algorithms not only represent breakthroughs in model structure but also conduct in-depth exploration in areas like feature representation and information fusion, fully demonstrating a continuous trend of methodological diversification and complication [14–17]. For example, Rhee et al. innovatively proposed a hybrid model architecture combining a Graph Convolutional Network (GCN) with a Relation Network. By effectively integrating gene expression profile data with known Protein-Protein Interaction (PPI) network information, they aimed to improve the classification accuracy and biological interpretability of breast cancer subtypes from a systems biology perspective [14].

Gao et al. designed and implemented a deep learning framework called DeepCC. This algorithm first calculates an enrichment score for each sample on predefined biological pathways based on gene expression profile data. It then feeds these pathway activity profiles as input features into a Fully Connected Neural Network for training, thereby achieving precise classification of cancer molecular subtypes [15].

Beykikhoshk and his team cleverly used an Attention Mechanism to dynamically calculate a personalized set of biomarkers and their corresponding weight scores for each test sample, specifically for the clinically confusable Luminal A and Luminal B breast cancer subtypes. They then performed subtype discrimination based on these weighted features, effectively enhancing the precision and interpretability of the classification [16]. Lee et al. further expanded the application of graph neural networks by developing a Pathway-associated Graph Attention Network model. This model demonstrated stable and excellent subtype classification performance on multiple independent breast cancer datasets, confirming its good generalization ability [17].

Furthermore, Yu et al., addressing the complexity and overlap between different breast cancer molecular subtypes, proposed a ”One-vs-Rest” strategy. This involves first screening for a set of differentially expressed genes significantly associated with each specific molecular subtype and then training binary classification models based on these specific gene sets. This approach aims to improve the identification ability and classification robustness for subtypes with ambiguous or confusable feature boundaries [12]. The aforementioned methods have fully tapped into the potential of machine learning in handling high-dimensional genomic data and complex pattern recognition, continuously driving improvements in classification accuracy.

1.3.3 Challenges and Prospects of Machine Learning in Breast Cancer Subtype Classification

Despite the encouraging progress of machine learning methods in breast cancer subtype classification research, there are still many challenges in their practical application and clinical translation. These challenges concern not only the performance of the models themselves but also the fit between the inherent characteristics of biomedical data and the needs of clinical practice. Specifically, the main issues are concentrated in the following areas:

- First is the prevalent **High-Dimension, Low Sample Size (HDLSS)** characteristic of breast cancer molecular data (especially genomic and transcriptomic data), where the number of features far exceeds the number of samples ($p \gg n$). This data structure makes it extremely easy for constructed models to fall into an overfitted state, with poor generalization ability on independent validation datasets, making them difficult to stably apply to new clinical samples [18, 19].
- Second, omics data from different research cohorts, using different experimental technology platforms, or with different data preprocessing pipelines, often have significant **batch effects and inherent platform differences**. Directly integrating such heterogeneous datasets to train a unified model often results in the model’s stability, reproducibility, and final prediction accuracy being affected by these non-biological factors.
- For supervised learning models, the **acquisition of subtype labels (ground truth) itself involves a certain degree of uncertainty and potential bias**. The molecular subtype labels of most clinical samples are indirectly inferred through methods like gene expression profiling (e.g., PAM50 test) or immunohistochemistry (IHC) staining. Factors such as different detection platforms, antibody choices, interpretation standards, and threshold settings can all lead to inconsistencies in subtype labels between different studies or even within the same study. This label noise undoubtedly poses a severe challenge to the robust training of supervised learning models.
- Although traditional machine learning models like SVM and Random Forest perform well on low-to-medium dimensional or relatively large sample size data, their performance can decline when the feature dimension increases dramatically beyond the sample size. The construction of their decision boundaries

can be significantly disturbed by noisy features in high-dimensional space. Therefore, implementing **effective feature selection, feature extraction, data dimensionality reduction, or model regularization strategies** to accurately extract the core signals that are truly related to subtypes from a massive number of original features and suppress noise is a crucial part of the model construction process.

- **Model interpretability is also a major challenge that needs to be addressed**: Clinicians and biologists are not only concerned with the accuracy of model predictions but also eager to understand the biological basis for the model's specific subtype discriminations, such as which specific genes, pathways, or molecular features play a key role in driving the classification decisions. However, many high-performance machine learning models, especially structurally complex deep neural networks, often have internal decision mechanisms that are like a "black box," making them difficult to interpret directly. This necessitates the use of additional interpretability analysis methods (such as SHAP, LIME) to enhance the model's transparency and clinical credibility [20].

These complex challenges have, to some extent, constrained the effective translation and widespread application of machine learning models in the field of breast cancer subtype classification. However, the application prospects of machine learning in the precise classification of breast cancer subtypes remain broad and full of potential. On one hand, its powerful data integration and pattern recognition capabilities help to break through the limitations of traditional classification based on single or few biomarkers, thus more comprehensively characterizing the molecular heterogeneity of breast cancer. On the other hand, machine learning also provides powerful computational tools for exploring and discovering potential new molecular subtypes or hidden complex biological patterns in the data.

With the continuous improvement of relevant algorithms, the accumulation of multi-omics data, and a deepening understanding of tumor biology, **machine learning is expected to play an increasingly pivotal role in promoting the precision of breast cancer molecular subtyping, the formulation of individualized diagnosis and treatment strategies, and the discovery of new therapeutic targets**.

1.4 Multi-omics Integration and Deep Learning: New Strategies for Modeling Complex Biological Features

1.4.1 Necessity and Challenges of Multi-omics Integrated Analysis

The development and progression of breast cancer is a complex biological process involving multiple stages and factors. Its molecular basis involves synergistic dysregulation and abnormal accumulation at multiple molecular levels, including the genome (e.g., gene mutations, copy number variations, chromosomal structural abnormalities), transcriptome (e.g., changes in the expression profiles of mRNA and non-coding RNA), epigenome (e.g., DNA methylation, histone modifications), proteome (e.g., protein expression, post-translational modifications, and interaction networks), and even metabolome [30].

Traditional analysis strategies based on single-omics data, while capable of revealing some biological information at a specific molecular level, often fail to comprehensively capture the complex interaction networks and synergistic regulatory mechanisms between different molecular levels, thereby limiting the depth of understanding of the overall biological characteristics of tumors. In view of this, **multi-omics data integration** has emerged, aiming to integrate heterogeneous molecular data from the same group of patient samples from a systems biology perspective, in order to reveal the complex biological landscape of tumor development and progression, and from which to discover more precise and robust biomarkers.

For example, for the same batch of tumor samples, researchers can simultaneously detect their DNA sequence variations and copy number changes, whole-genome DNA methylation profiles, the expression abundance of mRNA and miRNAs, and the expression levels and phosphorylation states of key proteins. These multi-dimensional data collectively construct a more comprehensive and personalized ”molecular fingerprint” for each tumor sample. However, their inherent high dimensionality, the heterogeneity between different omics data (such as differences in data type, scale, and distribution), and potential data missingness issues also pose higher requirements and challenges for subsequent computational analysis methods.

1.4.2 Application of Deep Learning in Multi-omics Integration and Representative Models

In recent years, artificial intelligence technologies represented by **Deep Learning** have made breakthrough progress and have rapidly penetrated various fields of biomedical research, opening up new strategic avenues for the effective integration and deep feature learning of complex high-dimensional multi-omics data [31, 32].

Compared to traditional machine learning algorithms that often rely on manually designed feature engineering steps, deep learning models (such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Graph Neural Networks (GNN), and Autoencoders (AE)) have the excellent ability to autonomously learn multi-level abstract feature representations from raw data, thereby capturing hidden linear and non-linear complex patterns in the data. This characteristic makes them highly suitable for handling the inherent complex associations and regulatory relationships between different molecular levels in multi-omics data. For example, Chaudhary et al. successfully developed a multi-omics integration model based on deep neural networks, which effectively fused gene expression profiles, DNA methylation profiles, and miRNA expression profiles of liver cancer patients. The experimental results showed that compared to models based on single-omics data, this integrated model could more accurately predict patient survival outcomes [33].

In the specific research direction of breast cancer subtype classification, multi-omics deep learning models for this task have also gradually emerged in recent years and have shown good potential.

Recently, Choi et al. designed and proposed a multi-omics integrated deep learning framework called ”**moBRCA-net**” specifically for the precise classification of breast cancer molecular subtypes [33]. This framework cleverly integrates three key types of omics information from the same patient cohort: gene expression data, DNA methylation data, and miRNA expression data. Its core innovation lies in equipping each type of omics data with a self-attention mechanism module, which can dynamically learn the relative importance or contribution weights of different molecular features (such as genes, methylation sites, miRNAs) in the subtype classification decision-making process. This allows for the adaptive highlighting of the most discriminative key biological signals and the suppression of irrelevant noise during the multi-modal information fusion process [33].

Through rigorous evaluation on public datasets, the moBRCA-net framework showed that, compared to baseline models trained on single-omics data, the effective fusion of multi-omics data significantly improved the accuracy and robustness of breast cancer subtype prediction. Similarly, Wang and collaborators developed a general-purpose multi-omics integration model called **MOGONET (Multi-Omics Graph cOnvolutional NETworks)** [34]. MOGONET uses a Graph Convolutional Network (GCN) to learn feature representations for each type of omics data separately, capturing its internal complex structural information. At the same time, through a designed cross-omics association learning mechanism (such as a view correlation discovery network), it effectively aggregates and aligns the feature representations learned from different omics data. This model has been successfully applied to multi-class molecular subtyping tasks for various tumor types, including breast cancer, and has shown superior performance to previous methods in several biomedical classification benchmark tests. These cutting-edge research examples fully demonstrate that, compared to traditional classification strategies that mainly rely on single data sources like gene expression profiles, well-designed deep learning models that effectively fuse multi-omics data can more comprehensively and deeply characterize the intrinsic biological properties of tumors, thereby significantly enhancing the overall performance of the classification models.

1.4.3 Advantages and Challenges of Multi-omics Deep Learning Methods

The application of multi-omics deep learning methods to breast cancer subtype classification tasks presents numerous significant advantages:

- First, by simultaneously considering information from different molecular levels (e.g., how specific gene mutations drive changes in downstream gene expression patterns, or how epigenetic modifications finely regulate the transcriptional activity of key genes), integrated models can capture more comprehensive and subtle molecular differences between subtypes, thereby effectively improving the biological relevance and clinical significance of the final classification results [30].
- The powerful non-linear fitting ability of deep learning models enables them to potentially uncover complex high-order interaction patterns that are difficult for traditional linear statistical methods or shallow learning models to effectively detect. For example, certain subtype-specific molecular feature combinations or synergistic regulatory networks may only become clearly apparent when jointly analyzing genomic variation, epigenetic modification, and transcriptomic expression data, and deep learning models are expected to automatically mine and learn such complex interaction effects from the data.
- Third, deep learning models (especially those based on architectures like Autoencoders) can construct a hierarchical feature representation learning mechanism to progressively abstract and compress raw high-dimensional, sparse multi-omics data into a lower-dimensional but more information-dense discriminative feature space. This, to some extent, effectively alleviates the ”curse of dimensionality” problem caused by data high-dimensionality, thereby effectively reducing the interference of inherent data noise and redundant information on model performance.
- With the introduction and integration of advanced technologies such as Autoencoders, Attention Mechanisms, and Graph Neural Networks, modern multi-omics deep learning models have made positive progress in automatically extracting key discriminative features and, to some extent, improving the interpretability of the model’s decision-making process. For example, the attention mechanism not only improves the predictive performance of the model, but the learned attention weight distribution can also intuitively indicate the relative importance of different omics features or sample regions in discriminating specific subtypes, providing valuable clues for researchers to identify potential biomarkers or understand the model’s decision logic [17].

Of course, applying multi-omics deep learning methods to complex biomedical problems like breast cancer subtype classification is not without its challenges. There are a series of significant issues to consider, such as the **inconsistency in scale and heterogeneity between different omics data** (e.g., gene expression data is usually more complete, while proteomic or metabolomic data may have more missing values or incomplete sample coverage); **how to effectively handle missing values and perform cross-modal data alignment**; the **high computational resources required for model training and the dependence on large-scale, high-quality labeled data**; and despite improvements, the still relatively complex **problem of model interpretability**.

Existing multi-modal integration methods, such as multi-kernel learning and canonical correlation analysis (CCA), often rely on manual feature selection or can only capture linear associations, lacking the ability to automatically extract and fuse complex non-linear relationships from different omics data [35], which limits their potential in in-depth biological mechanism discovery. Overall, **the deep learning strategy of integrating multi-source heterogeneous data undoubtedly provides a more comprehensive and in-depth analytical perspective and more powerful computational modeling tools for breast cancer subtype classification research**. In related published studies, well-designed multi-omics integration models often show significant superiority over traditional single-omics models in terms of classification accuracy, model robustness, and generalization ability to new samples [17].

By further optimizing the network architecture design of deep learning models, innovating multi-modal data fusion mechanisms, and more closely combining domain biological knowledge to effectively constrain and guide the model, we can expect to accurately identify the key molecular features and core regulatory networks driving the development of different breast cancer subtypes from massive, complex multi-omics data. This is expected to push the research of fine-grained molecular subtyping of breast cancer to new heights and provide strong support for achieving true personalized precision medicine.

1.5 High-Dimension Low-Sample-Size (HDLSS) Problem and Data Augmentation Techniques: Generalization and Biological Signal Extraction

1.5.1 The High-Dimension Low-Sample-Size (HDLSS) Problem: Curse of Dimensionality and Challenges

In the field of biomedical data analysis, especially in studies involving multi-omics data such as genomics, transcriptomics, and epigenomics, the **High-Dimension, Low-Sample-Size (HDLSS)** phenomenon constitutes a pervasive and highly challenging problem. This issue is particularly prominent and intractable in the context of breast cancer multi-omics research.

Typical transcriptomic sequencing data or whole-genome DNA methylation array data often have feature dimensions (e.g., the number of detected genes or CpG sites) reaching tens or even hundreds of thousands. However, due to factors such as the difficulty in obtaining clinical samples, research costs, and ethical considerations, the number of patient samples available for analysis (i.e., the number of observations) is usually only in the dozens to hundreds.

This extremely unbalanced data structure, where the ”number of variables far exceeds the number of observations” ($p \gg n$), makes it very easy for machine learning models to overfit the specific noise and random fluctuations in the training data, rather than learning generalizable, true biological patterns. The direct consequence is that the model performs excellently on the training set but has poor generalization ability on unseen data, a phenomenon known as the **”curse of dimensionality”** [18]. Therefore, in the HDLSS context, how to effectively improve the generalization performance of the model and extract robust and reliable biological signals from it is a core issue of such research.

1.5.2 Feature Selection and Data Dimensionality Reduction Strategies

Facing the challenge of high-dimensional data, a direct and widely used strategy is to carefully select and streamline the original feature set before building a predictive model. The aim is to effectively reduce the data dimensionality by removing redundant and irrelevant features, thereby focusing on more informative biological signals, while also reducing the model’s computational burden and potentially enhancing its interpretability.

Over the past few decades, researchers have developed a large number of feature selection and feature extraction methods for high-dimensional biomedical data such as gene expression data. These methods can be broadly classified into three categories: filter, wrapper, and embedded methods. Common examples include: univariate filter methods based on statistical tests (e.g., t-test, ANOVA, chi-squared test) to initially screen for genes with significantly different expression levels between different subtypes; embedded methods based on sparse learning theory, such as LASSO (Least Absolute Shrinkage and Selection Operator) regression or Ridge Regression and their variants (e.g., Elastic Net), which can automatically select important features and shrink the coefficients of other features to zero or near zero during model training by introducing L1 or L2 norm penalty terms in the loss function; and unsupervised data dimensionality reduction methods, such as Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), or Uniform Manifold Approximation and Projection (UMAP), which **aim to project the original high-dimensional data into a lower-dimensional subspace while preserving as much of the original structure or variance information as possible** [18].

Practice has shown that in breast cancer subtype classification tasks, appropriate feature selection or dimensionality reduction preprocessing steps can often significantly improve the performance, stability, and biological interpretability of the subsequent classification model. For example, the well-known PAM50 gene set itself is a set of 50 representative genes selected from thousands of candidate genes. Using these pre-selected landmark genes instead of the whole-genome expression profile to train a classification model can not only effectively reduce the interference of data noise and enhance the biological significance of the model but also reduce the cost and complexity of subsequent clinical testing.

1.5.3 Data Augmentation Techniques: Application of Generative Models (GANs, VAEs)

In addition to addressing the HDLSS problem from the feature level, another important strategy aimed at improving model generalization is **data augmentation**, which involves ”artificially” expanding the effective size of the training dataset through various technical means while maintaining the original data distribution characteristics, thereby providing the model with richer and more diverse learning materials.

In the field of computer vision, data augmentation techniques based on geometric or optical transformations such as image rotation, flipping, cropping, and color jittering have been widely proven to significantly improve the robustness and generalization ability of deep learning models. However, for structured numerical high-dimensional data such as gene expression profiles and DNA methylation profiles, traditional data augmentation methods based on simple geometric transformations or noise injection are often not directly applicable or have limited effects.

In recent years, the rise of deep generative models, represented by **Generative Adversarial Networks (GANs)**, has provided promising new ideas for the task of augmenting such data. A GAN consists of a Generator and a Discriminator, which are trained iteratively through an adversarial ”zero-sum game” process: the Generator strives to learn the intrinsic distribution of real data and produce ”pseudo-samples” that are as realistic as possible, while the Discriminator tries to distinguish between real samples and the pseudo-samples produced by the Generator. Through this dynamic competition and co-evolution, ideally, the Generator can capture the high-dimensional complex distribution of the real data and generate new synthetic samples with high realism and diversity. Researchers have actively tried to apply GANs and their variants to the task of augmenting gene expression data. For example, Chaudhari et al. proposed an improved GAN architecture (**MG-GAN**), which introduces a Gaussian noise layer in the Generator's network structure for perturbation. Experiments have shown that this method can successfully simulate synthetic data with a distribution highly similar to real cancer gene expression data [31].

Kwon et al. found in practice that directly using the complete expression matrix containing tens of thousands of genes to train a GAN model often yielded unsatisfactory results, with the quality of the synthetic data being difficult to guarantee. To address this, they proposed a strategy of first screening for a more information-rich subset of significant genes through feature selection methods, and then training the GAN model and generating data only for this selected gene subset (rather than the entire genome). This effectively improved the quality and biological relevance of the synthetic data [32].

Ahmed et al. further extended the idea of data augmentation to the multi-omics data scenario by developing an integrated generative model called **omicsGAN**. This model can simultaneously feed gene expression data and another related omics data (e.g., DNA methylation data) as joint inputs into the GAN framework, aiming to learn and capture the intrinsic association structures between different omics modalities and generate more realistic and comprehensive multi-omics synthetic samples based on them [36]. These innovative data augmentation methods can produce synthetic data that can be effectively used to expand the original training dataset, thereby alleviating the problem of insufficient model training or overfitting caused by a limited number of original training samples.

Several studies have preliminarily shown that supplementing the training of classification models with additional synthetic samples generated by techniques such as GANs can indeed achieve better predictive performance on independent test datasets, especially in scenarios where the original training sample size is extremely limited, where the performance improvement may be even more significant [32].

In addition to GANs, other types of deep generative models, such as **Variational Autoencoders (VAEs)**, have also been explored for augmenting biomedical data. Furthermore, some researchers have tried to apply ideas based on geometric space interpolation or perturbation to high-dimensional data augmentation, for example, by creating new synthetic samples located between real samples or in their vicinity through linear or non-linear interpolation in the feature space, or by applying controlled random perturbations in the neighborhood of real samples, to enrich the diversity and coverage of the training data [16].

However, it should be emphasized that the implementation of any data augmentation strategy should be carefully evaluated to avoid introducing systematic biases that could mislead model learning. The choice of generative model, parameter tuning, and quality control of the generated samples directly determine the final effectiveness of data augmentation. If the generated synthetic data lacks sufficient biological realism (e.g., fails to accurately reflect true gene co-expression patterns, pathway activation states, or subtype-specific features), then using it for model training may instead have a negative impact on model performance. Therefore, **a detailed evaluation of the biological plausibility of the generated samples (e.g., whether key gene correlations and pathway features are preserved) is an indispensable key step when applying data augmentation techniques to biomedical data analysis**.

1.5.4 Comprehensive Strategies for Improving Generalization and Biological Signal Extraction

When dealing with high-dimensional, low-sample-size (HDLSS) biomedical data, improving the generalization ability of machine learning models is key to their potential clinical utility. The core goal is not just to increase the predictive accuracy of the model on unseen data, but more deeply, to ensure that the constructed model can learn and capture the true biological signals that reflect the underlying mechanisms of the disease, rather than merely fitting the random noise or specific patterns of a particular dataset. To achieve this goal, a comprehensive, multi-pronged approach is usually required, including the following aspects:

1. **Strict Regularization Techniques and Robust Cross-Validation Schemes**: During the model training phase, various effective regularization techniques, such as L1/L2 weight penalties, Dropout (for neural networks), and Early Stopping, should be widely used to constrain model complexity and prevent

overfitting. At the same time, robust cross-validation strategies such as k-fold cross-validation, Leave-One-Out Cross-Validation (LOOCV), or repeated random sub-sampling validation must be employed to objectively and unbiasedly evaluate the generalization performance of the model [18].

2. **Integration of Multiple Datasets and Application of Transfer Learning**: When conditions permit, actively attempt to integrate datasets from different research cohorts, different geographic populations, or using different experimental technology platforms for joint analysis. Alternatively, use transfer learning strategies to transfer the knowledge from models pre-trained on large-scale related datasets (such as pan-cancer datasets) to target tasks with relatively small sample sizes, in order to improve the model's adaptability and robustness to different data distributions and potential batch effects.

3. **Incorporating Biological Prior Knowledge for Guidance and Constraint**: Make full use of the accumulated biological prior knowledge in the field, such as known pathogenic genes, key signaling pathways, gene regulatory networks, or protein-protein interaction networks, to guide the feature selection process (e.g., prioritizing genes known to be closely related to breast cancer development as candidate features), or directly incorporate these biological structural information into the model architecture design (e.g., building graph neural network models based on biological networks). This knowledge-guided learning approach helps to guide the model to focus on signals with real biological significance, rather than accidental or spurious correlations that may exist in the data, thereby enhancing the interpretability and biological meaning of the model.

Overall, **the HDLSS problem is undoubtedly a major and unavoidable challenge in the research field of breast cancer subtype classification and even the entire field of biomedical data analysis, but it has also become the core driving force for the continuous innovation of related computational methods and analysis strategies**. By comprehensively applying a series of technical means such as feature selection, data dimensionality reduction, data augmentation, model regularization, and knowledge-guided learning, researchers are committed to gradually improving the robustness, generalization performance, and biological interpretability of machine learning models on such data. More importantly, the optimization and application of these technical means are not only for improving the predictive accuracy of classification models. The more profound significance lies in promoting the model to effectively extract key signals with real biological meaning from complex high-dimensional data, such as accurately identifying the core gene modules, key molecular pathways, or specific biomarker combinations that drive the occurrence, development, metastasis, or treatment resistance of different breast cancer subtypes. This not only directly contributes to the improvement of classification model performance and clinical translation but also has the potential to uncover key driver gene modules or molecular regulatory pathways related to different breast cancer subtypes, providing new clues and perspectives for in-depth understanding of the biological mechanisms of breast cancer. Therefore, in the process of continuous development in this field full of challenges and opportunities, how to achieve a delicate balance between model complexity, data dimensionality, and available sample size, and how to fully mine the potential information of high-dimensional data without falling into the trap of overfitting, will undoubtedly continue to be the core focus of researchers in this field.

In summary, the heterogeneity of breast cancer highlights the importance of efficient, interpretable multi-omics subtyping models. The TCGA-BRCA data provides an ideal foundation for such research. This study aims to develop a breast cancer molecular subtyping framework with high accuracy, strong generalization ability, and good interpretability by integrating deep learning and network biology methods based on TCGA-BRCA multi-omics data, promoting its application in pathogenic mechanism analysis and clinical decision support.

1.6 Scope, Core Objectives, and Structure of This Report

Given the preceding systematic review of breast cancer molecular heterogeneity, subtype classification evolution, mainstream machine learning methods, and the challenges they face in handling high-dimension, low-sample-size (HDLSS) multi-omics data, the research work of this report will focus on a specific scope and pursue clear objectives.

This report aims to conduct a comprehensive, comparative analysis and evaluation of the practical effectiveness of several mainstream machine learning models—specifically, **LightGBM, XGBoost, Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), and Support Vector Machine (SVM)**—in the task of cancer subtype classification, particularly in the discrimination of breast cancer molecular subtypes. This analysis will primarily rely on a series of carefully designed computational experiments conducted by our research team on a specific breast cancer dataset and their resulting empirical findings. At the same time, to ensure the depth and breadth of the analysis, this report will also extensively integrate and reference existing published academic research literature (such as [21] to [41], see the bibliography for a detailed list), drawing insightful perspectives from them to provide necessary in-depth interpretation, theoretical support, and background supplementation for the experimental findings of this study.

The core research focus of this report will be on the **precise subtype classification of breast cancer according to the internationally recognized PAM50 molecular subtyping standard**. At the data level, we will primarily utilize patient genomic data, with a particular emphasis on high-throughput **gene expression profiles**. Meanwhile, considering the potential key role of Gene Copy Number variation (GCN) in tumor development and subtype characteristic formation, this report will also carefully evaluate and discuss the potential benefits and implementation pathways of incorporating **Gene Copy Number (GCN) data** into the classification models.

Centered around this core focus, this report will delve into and meticulously analyze the following key aspects:

1. **Comparative analysis of performance metrics for each machine learning model**: Systematically compare the key performance evaluation metrics of LightGBM, XGBoost, MLP, CNN, and SVM in the PAM50 subtype classification task, such as Accuracy, Precision, Recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC), and conduct detailed error type analysis using confusion matrices.

2. **Hyperparameter optimization strategies and their impact on model performance**: Detail the hyperparameter optimization strategies and processes adopted for each model, and deeply analyze the sensitivity and specific impact of different hyperparameter configurations on the final classification performance of the model.

3. **Specific methodological considerations for genomic data**: Focus on the unique challenges faced when processing high-dimensional, heterogeneous genomic data (especially HDLSS gene expression profiles and GCN data), such as feature selection, data dimensionality reduction, batch effect correction, and model regularization, as well as the specific methodological countermeasures taken in this study to address these challenges.

4. **Relevance and potential significance of the research findings in a broader biological and clinical context**: Interpret the experimental results of this study within the macroscopic context of current breast cancer molecular biology research progress and clinical practice needs, exploring the potential biological insights they may reveal, their implications for existing classification systems,

and their potential future application value in assisting clinical diagnosis, prognostic evaluation, or treatment decision-making.

To clearly present an overview of the machine learning models examined in this report, the following table (Table 1) is provided:

Table 1: Overview of Machine Learning Models Analyzed in This Study

Model Name	Algorithm Type/Family	Key Features & Potential Advantages in Genomic Data Classification	Relevant Literature
LightGBM	Gradient Boosting Decision Tree (GBDT) Ensemble	Fast training speed, relatively low memory consumption, supports efficient parallel learning, suitable for handling large-scale high-dimensional data.	[37]
XGBoost	Gradient Boosting Decision Tree (GBDT) Ensemble	Built-in advanced regularization mechanisms (e.g., L1/L2 penalties) effectively prevent overfitting, can automatically handle sparse data and missing values, highly scalable and flexible, supports custom objective functions.	[37]
MLP	Feedforward Neural Network	Powerful ability to fit complex non-linear relationships, flexible network structure design, can learn hierarchical feature representations through multiple layers of abstraction.	[38]
CNN	Convolutional Neural Network	Excels at learning local spatial or sequential patterns in data (e.g., motifs in gene expression sequences), features weight sharing and translation invariance, performs well on grid-like or 1D sequential data (with appropriately transformed gene expression profiles).	[38]
SVM	Kernel-based Classifier	Robust performance in high-dimensional feature spaces, especially when the number of features is much larger than the number of samples (HDLSS), can handle complex non-linearly separable problems through the kernel trick, has a clear decision boundary, and generally good generalization ability.	[37]

2 Data Sources and Preprocessing

The analysis in this study is based on publicly available data from The Cancer Genome Atlas (TCGA) Breast Cancer (BRCA) project. The TCGA project, through the integration of multi-platform and multi-omics data, provides a valuable resource for understanding the molecular basis of cancer [24]. Data acquisition and preprocessing are crucial for building robust machine learning models, and this section will detail this process.

2.1 Data Sources

This study primarily integrated the following two types of data from the TCGA-BRCA project, both obtained from the UCSC Xena Hub (tcga.xenahubs.net) public database:

1. **Gene Copy Number Variation (CNV) Data:** Gene copy number variation is a common structural variation in cancer genomes, involving the amplification or deletion of DNA segments, which can lead to

the activation of oncogenes or the inactivation of tumor suppressor genes, thereby driving tumorigenesis and progression. The dataset used in this study is gene-level copy number estimates processed and thresholded by the GISTIC2 algorithm. GISTIC2 (Genomic Identification of Significant Targets in Cancer 2.0) is a widely recognized algorithm for identifying statistically significant recurrent copy number variation regions (including amplifications and deletions) in a large number of cancer samples [27]. The output of this algorithm is typically discretized copy number values (e.g., -2 for homozygous deletion, -1 for heterozygous deletion, 0 for normal diploid, 1 for low-level amplification/gain, 2 for high-level amplification), which helps to reduce noise in the raw CNV data and provide more interpretable features for subsequent machine learning analysis. The raw CNV data is usually stored in a gene (rows) x sample (columns) matrix format.

2. **Clinical Phenotype Data:** This dataset contains detailed clinical annotation information for each sample in the TCGA-BRCA project. This information includes, but is not limited to, patient demographic characteristics, tumor pathological features (such as grade and stage), treatment history, and key molecular subtyping results. In this study, the most critical clinical information is the breast cancer molecular subtype classification based on the PAM50 gene expression profile. The PAM50 gene set contains 50 selected genes whose expression patterns can reliably classify breast cancer into five major ”intrinsic subtypes”: Luminal A, Luminal B, HER2-enriched, Basal-like, and Normal-like [3,5]. These subtypes show significant differences in prognosis, response to treatment, and underlying biological driver mechanisms, making PAM50 subtyping an important standard in breast cancer clinical research and practice. The ”PAM50_mRNA_nature2012” column in the clinical data was used as the ground truth for the classification task in this study.

2.2 Data Preprocessing Pipeline

To construct a standardized dataset suitable for machine learning models, a series of preprocessing steps were applied to the raw data:

1. **Initial Data Loading and CNV Data Transposition:** First, the CNV data matrix and clinical phenotype data were loaded separately. The original CNV data matrix was transposed so that rows represent sample IDs and columns represent genes (i.e., features), a format more compliant with the input requirements of most machine learning algorithms.

2. **Tumor Sample Filtering and Subtype Label Extraction:** TCGA sample IDs have a specific structure, with the 14th-15th characters encoding the sample type. This study selected only samples representing primary solid tumors (usually encoded as ’01’). For these filtered tumor samples, their corresponding ”PAM50_mRNA_nature2012” subtype labels were extracted from the clinical data. During this process, samples with missing (e.g., NA, empty string) or invalid subtype labels were removed.

3. **Handling of Rare Subtypes:** In multi-class classification tasks, the number of samples in some classes may be much smaller than in others, leading to a class imbalance problem, which can cause the model to be biased towards the majority class and have difficulty learning the features of the minority classes. To mitigate this issue, this study removed subtypes with very few samples in the dataset (specifically, a threshold of fewer than 7 samples). Although this processing may result in some information loss, it helps to improve the stability of the model and its classification performance on the major subtypes.

4. **Data Merging and Initial Cleaning:** The processed CNV feature data and the corresponding subtype label data were merged based on a unique sample ID using an inner join. This step ensured

that every sample in the dataset had both a CNV feature profile and a clear subtype assignment. In the merged data, any remaining missing values in the CNV features (though rare in GISTIC2 thresholded data) were filled with 0. The value 0 was chosen for imputation based on the nature of CNV data, where 0 typically represents a normal diploid state or no significant copy number change, and can be considered a neutral or baseline state.

5. **Feature Selection and Dimensionality Reduction:** Genomic data typically has extremely high dimensionality (thousands of genes) and a relatively small sample size, the so-called ”High-Dimension, Low-Sample-Size” (HDLSS) problem [18, 38]. This data structure is highly prone to model overfitting, which reduces its generalization ability. To address this challenge, this study implemented the following two-step feature selection strategy:

   - **Low-Variance Feature Removal:** First, the variance of each gene feature across all samples was calculated. Features with a variance of 0 (i.e., having the exact same value across all samples) contain no information for distinguishing between different subtypes and were therefore removed directly.
   - **High-Variance Feature Pre-filtering:** After removing zero-variance features, the remaining gene features were sorted in descending order of their variance. The top 150 genes with the highest variance were selected as the final feature set for the models. Variance is often used as a simple indicator of a feature’s information content; genes with higher variability are more likely to be associated with differences in phenotypes (such as cancer subtypes). Reducing the number of features from tens of thousands to 150 was intended to significantly decrease the computational complexity of model training, reduce noise and redundant information, and focus the analysis on the signals most likely to be biologically meaningful, thereby improving the model’s learning efficiency and potential generalization ability. The choice of 150 as the feature count threshold was based on experience and computational resource considerations, aiming to balance information retention and model complexity.

6. **Feature Name Normalization:** Many machine learning packages (especially in the R environment) have specific requirements for the format of column names (i.e., feature names), such as not allowing spaces, special characters, or starting with a number. To ensure compatibility, the `make.names()` function was used to standardize the names of all 150 selected gene features, generating unique column names that conform to R language specifications.

7. **Data Splitting:** Finally, the dataset obtained after all the above preprocessing steps, containing 150 CNV features and subtype labels, was split into 60% (training set), 20% (validation set), and 20% (test set). The splitting process used a stratified sampling strategy to ensure that the proportion of samples for different cancer subtypes in each data subset was consistent with the proportion in the original complete dataset. The training set was used for model parameter learning; the validation set was used for hyperparameter tuning and monitoring the training process to prevent overfitting (e.g., through early stopping); and the test set served as the final, independent evaluation set to measure the true generalization performance of the trained model on unseen data.

Through the above detailed data source confirmation and multi-step preprocessing, we aimed to build a high-quality, information-rich, and dimensionally appropriate dataset, laying a solid foundation for the subsequent machine learning model training and cancer subtype classification evaluation.

3 Machine Learning Model Overview

This study selected a representative set of machine learning models, covering different paradigms such as ensemble learning, deep learning, and kernel methods, to comprehensively evaluate their performance in the task of breast cancer PAM50 subtype classification based on gene copy number variation data. The following sections provide a detailed introduction to each selected model, explaining its core principles, key technologies, and its applicability and considerations in the context of genomic data analysis.

3.1 LightGBM (Light Gradient Boosting Machine)

LightGBM is a highly efficient ensemble learning algorithm based on Gradient Boosting Decision Trees (GBDT) [37]. GBDT is a powerful ensemble method whose core idea is to iteratively train a series of weak learners (usually decision trees). In each iteration, a new decision tree is trained to fit the residuals of the predictions accumulated from all previous trees (i.e., the difference between the true value and the current ensemble model's prediction). In this way, the model gradually reduces errors and improves overall predictive performance. LightGBM introduces several innovative techniques on top of traditional GBDT, significantly improving training speed and memory efficiency, making it particularly suitable for handling large-scale and high-dimensional datasets like genomic data.

Its key technical features include:

1. **Gradient-based One-Side Sampling (GOSS):** In traditional GBDT, all training samples are treated equally when calculating information gain (for selecting the best split point). However, the GOSS algorithm posits that samples with larger gradients (i.e., samples for which the model's current prediction error is large) contribute more to the information gain, as they are the ”hard” samples that the model has not yet learned well. Therefore, the GOSS strategy is to retain all samples with large gradients and perform random sampling on the samples with small gradients. In this way, GOSS significantly reduces the number of samples to be considered while maintaining model accuracy, thereby lowering computational complexity and accelerating the training process.

2. **Exclusive Feature Bundling (EFB):** In high-dimensional sparse datasets (for example, in genomic data, many genes' copy number variations may only occur in a few samples, or some genes may have mutually exclusive variation patterns), many features are mutually exclusive, meaning they rarely take non-zero values simultaneously. The EFB algorithm leverages this property to bundle these mutually exclusive or nearly mutually exclusive features into a single ”feature bundle”. By constructing a graph where nodes represent features and edges represent conflicts between features (i.e., whether they frequently take non-zero values at the same time), EFB can merge features with few conflicts. This can effectively reduce the number of features without significant information loss, thereby reducing the computational cost of building feature histograms (see next point).

3. **Leaf-wise tree growth with depth limitation:** Traditional GBDT algorithms (like early versions of XGBoost) typically use a level-wise or depth-wise growth strategy, where all leaf nodes at the same level are split simultaneously. In contrast, LightGBM defaults to a leaf-wise growth strategy. It selects the leaf node that will yield the maximum information gain after splitting from all current leaf nodes. This strategy usually produces a lower training loss for the same number of splits compared to the level-wise strategy, potentially leading to higher model accuracy. However, unconstrained leaf-wise growth can easily lead to the generation of very deep, asymmetric decision trees, which can cause overfitting on datasets with a small number of samples. Therefore, LightGBM is usually used in conjunction with a

maximum depth limit (`max_depth` parameter) to control the complexity of the tree, striking a balance between accuracy and overfitting.

4. **Histogram-based Algorithm for Decision Tree Learning:** To speed up the process of finding the best split point, LightGBM discretizes continuous floating-point feature values into a fixed number of bins (e.g., 256) and builds feature histograms based on these bins. When calculating information gain and selecting a split point, the algorithm only needs to iterate over these discrete bin boundaries, rather than over the exact feature value of each sample. This greatly reduces the number of candidate split points, significantly accelerates the training process, and effectively reduces memory consumption because it is no longer necessary to store sorted feature values.

In addition to the core technologies mentioned above, LightGBM also supports direct handling of categorical features, parallel learning (feature parallel, data parallel, voting parallel), and efficient cache optimization. These features enable LightGBM to exhibit significant speed and efficiency advantages while maintaining or even surpassing the performance of other GBDT algorithms when processing genomic datasets containing tens of thousands of gene features and thousands of samples. However, like other GBDT models, its performance is highly dependent on careful hyperparameter tuning, and for datasets with very small sample sizes or extremely high noise, the risk of overfitting still needs to be guarded against.

3.2 XGBoost (Extreme Gradient Boosting)

XGBoost (Extreme Gradient Boosting) is another very popular and high-performance implementation of the Gradient Boosting Decision Tree (GBDT) algorithm, developed under the leadership of Dr. Tianqi Chen [37, 39, 40]. It builds on the GBDT algorithm with in-depth theoretical analysis and systematic engineering optimization, leading to its great success in machine learning competitions and various practical applications, including bioinformatics.

The core features and advantages of XGBoost lie in its comprehensive consideration of model complexity, regularization, computational efficiency, and flexibility:

1. **Regularized Learning Objective:** When defining the optimization objective function, XGBoost explicitly adds a regularization term in addition to the traditional loss function (which measures the difference between model predictions and true values) to control the model's complexity and prevent overfitting. Its objective function can be expressed as:

$$\mathrm{Obj}(\Theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{1}$$

where $l(y_i, \hat{y}_i)$ is the loss function for the i-th sample, and $\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$ is the ensemble prediction of K trees.

$\Omega(f_k)$ is the complexity penalty term for the k-th tree, usually defined as:

$$\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda \sum_{j=1}^{T_k} w_j^2 + \alpha \sum_{j=1}^{T_k} |w_j| \tag{2}$$

Here, T_k is the number of leaf nodes in the k-th tree, and w_j is the weight of the j-th leaf node (i.e., the output value of the leaf node). The γ parameter controls the number of leaf nodes (penalizing tree complexity), λ is the L2 regularization coefficient (making leaf weights smoother), and α is the L1 regularization coefficient (which can make leaf weights sparse, acting as a form of feature selection, although not directly on the original features in a tree model). This regularized objective function allows XGBoost to better balance the model's fitting ability and generalization ability during optimization.

2. **Sparsity-aware Split Finding:** Many real-world datasets (including gene expression or variation data in genomics) may contain missing values or exhibit sparsity. XGBoost can automatically handle these situations without requiring explicit imputation from the user. When splitting a node during tree construction, the algorithm learns a "default direction" for missing values, meaning that when a sample is missing a value for a feature, it is automatically assigned to either the left or right child node. This default direction is determined by comparing the information gain that can be achieved by assigning the missing values to the left and right child nodes, respectively.

3. **Approximate Greedy Algorithm and Weighted Quantile Sketch:** For very large datasets, the exact greedy algorithm (i.e., iterating over all possible split points for all features to find the optimal split) is computationally too expensive. XGBoost uses an approximate algorithm: it first proposes a set of candidate split points based on the percentiles of the feature values and then finds the optimal split only among these candidates. To more effectively handle weighted samples (for example, in some custom loss functions or weighted processes of gradient boosting), XGBoost uses a weighted quantile sketch algorithm to generate these candidate split points, ensuring they can well reflect the distribution of the weighted data.

4. **Efficient Parallel and Distributed Computing:** XGBoost was designed with computational efficiency in mind. On a single machine, it can use multi-core CPUs for parallel processing, especially at the feature level during tree construction (e.g., the computation for finding the best split point for different features can be parallelized). In addition, XGBoost supports running in distributed computing environments (like Hadoop, Spark), enabling it to handle larger-scale datasets.

5. **Cache-aware Access and Out-of-core Computation:** To further improve performance, XGBoost has optimized the way data is read from memory and CPU cache to reduce cache misses. At the same time, it supports out-of-core computation, allowing it to process massive datasets that cannot be fully loaded into main memory by storing data in blocks on the hard drive and reading them into memory as needed for processing.

6. **Built-in Cross-Validation and Flexibility:** XGBoost provides a built-in cross-validation function (`xgb.cv`), which is convenient for users to perform hyperparameter tuning and model evaluation. In addition, it allows users to define custom loss functions and evaluation metrics, offering high flexibility to adapt to different task requirements.

Due to its powerful predictive performance, effective control of overfitting, good adaptability to various data types, and efficient computational implementation, XGBoost has become a very popular and powerful benchmark model in the field of genomic data analysis and cancer subtype classification.

3.3 Multilayer Perceptron (MLP)

A Multilayer Perceptron (MLP) is one of the most basic and widely used types of feedforward artificial neural networks (ANN). It consists of at least three layers of neurons (nodes): an input layer, one or more hidden layers, and an output layer. An MLP learns the complex non-linear mapping relationship between input data and output targets to perform classification or regression.

Its core components and working principles are as follows:

1. **Network Structure:**

- **Input Layer:** Receives the raw feature data. Each neuron corresponds to one feature dimension of the input data. For genomic data, this is typically the gene expression value, copy number variation value, or other molecular features of each sample.
- **Hidden Layers:** An MLP can contain one or more hidden layers. Each neuron in a hidden layer is connected to all neurons in the previous layer (the input layer or the previous hidden layer) through weighted connections (synaptic weights). Each neuron calculates the weighted sum of all its inputs, adds a bias term, and then transforms the result through a non-linear activation function. Common activation functions include the Sigmoid function (compresses output to 0-1, often used in binary classification output layers or early networks), the hyperbolic tangent function (Tanh, compresses output to -1 to 1), and the Rectified Linear Unit (ReLU, i.e.,

  $$f(x) = \max(0, x) \tag{3}$$

  which has become the most commonly used activation function in modern neural networks because it alleviates the vanishing gradient problem and is computationally efficient). The number of hidden layers and the number of neurons per layer together determine the capacity and complexity of the network.
- **Output Layer:** The structure and activation function of the output layer depend on the specific task. For multi-class classification tasks (like cancer subtype classification), the output layer usually contains the same number of neurons as the number of classes and uses the Softmax activation function. The Softmax function converts the output of each neuron into the probability of the corresponding class (the sum of all probabilities is 1).

2. **Learning Process (Backpropagation Algorithm):** The learning process of an MLP (i.e., parameter optimization) is usually accomplished through the backpropagation algorithm combined with gradient descent (or its variants, such as Adam, RMSprop).

   - **Forward Propagation:** The input data propagates forward from the input layer to the output layer, calculating the model's predicted output.
   - **Loss Calculation:** A predefined loss function is used to measure the difference between the model's prediction and the true label. For multi-class classification tasks, the common loss function is categorical cross-entropy.
   - **Backward Propagation:** The gradient of the loss function with respect to all weights and biases in the network is calculated. This process starts from the output layer and calculates the gradients layer by layer backward using the chain rule.
   - **Weight Update:** The calculated gradients are used to update the weights and biases in the network through a gradient descent optimization algorithm, with the goal of minimizing the loss function. This process is iterated over the entire training dataset (usually for multiple epochs) until the model converges or a preset stopping condition is met.

3. **Non-linear Mapping and Feature Representation Learning:** Due to the use of non-linear activation functions in the hidden layers, an MLP can theoretically approximate any complex continuous function (Universal Approximation Theorem), which enables it to learn highly non-linear patterns and complex interactions between features in the input data. In a multi-layer structure, each layer can be seen as a higher-level, more abstract representation of the features from the previous layer's output. For example, the first hidden layer might learn some low-level features, and subsequent hidden layers combine these low-level features to learn higher-level, more complex feature representations. This ability for hierarchical feature learning is a core advantage of deep learning models.

In the context of genomic data and cancer subtype classification, an MLP typically takes the gene expression profile (or other molecular feature vectors) of each sample directly as input [38]. Although MLP has powerful modeling capabilities, it also faces significant challenges when dealing with HDLSS (High-Dimension, Low-Sample-Size) genomic data:

- **Risk of Overfitting:** MLP models, especially those with many layers or many units per layer, often contain a large number of trainable parameters. When trained on genomic data with a relatively small number of samples, the model can easily overfit the specific noise and patterns of the training set, leading to poor generalization ability on independent test data. Therefore, strong regularization techniques are key to the successful application of MLP on genomic data. Common regularization methods include Dropout (randomly deactivating a portion of neurons during training), L1/L2 weight decay (adding a penalty term for weights to the loss function), and Early Stopping (monitoring the performance on a validation set and stopping training when performance no longer improves).
- **Equal Treatment of Input Features and Lack of Structural Information:** A standard MLP treats all input features as independent dimensions and processes them through fully connected layers. This structure may not directly utilize known biological relationships or structural information between genes (such as upstream/downstream relationships in signaling pathways, gene co-expression modules, or proximity on the genome), unless this information is pre-encoded into the features or captured through more complex network architectures (such as introducing attention mechanisms or graph neural networks).
- **Hyperparameter Sensitivity and Training Complexity:** The performance of an MLP is highly dependent on the careful selection and tuning of numerous hyperparameters, including the network architecture (number of hidden layers, number of neurons per layer), choice of activation function, choice of optimizer and its parameters (like learning rate, momentum), batch size, and regularization strategies and parameters. Finding the optimal combination of hyperparameters often requires a large amount of experimentation and computational resources.

Despite these challenges, MLP, due to its relatively simple concept, flexible implementation, and powerful non-linear modeling capabilities, is still widely used in bioinformatics as a baseline model for classification and prediction tasks, or as an important component in more complex deep learning architectures (such as autoencoders for feature dimensionality reduction, or graph neural networks for integrating network information) [41].

3.4 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a special class of deep learning models whose architecture is particularly well-suited for processing data with a grid-like topology, such as images (2D grids), videos (3D grids, including a time series), or sequential data (1D grids). Although CNNs are initially famous for their breakthrough achievements in the field of computer vision, their core ideas and components have been successfully extended and applied to process other types of data, including 1D gene expression profiles for cancer subtype classification.

The core components and working principles of a 1D CNN for processing sequential data like gene expression profiles include:

1. **1D Convolutional Layer:** This is the core of a 1D CNN. Similar to the 2D convolutional layer for processing images, the 1D convolutional layer contains a set of learnable filters (or kernels). Each filter

is a small weight vector (e.g., with a length of 3, 5, or 7 genes), which slides (convolves) over the input 1D gene expression sequence. At each position, the filter is element-wise multiplied with the segment of the input sequence it covers and summed (dot product), and a bias term is added. This process aims to detect specific local patterns or features in the input sequence. For example, in a gene expression profile, a 1D convolutional kernel might learn to recognize a small segment of co-upregulated or co-downregulated genes, or a specific expression ”shape” or ”motif”. By using multiple different filters, a convolutional layer can learn various types of local features in parallel. The two key properties of a CNN are:

- **Weight Sharing:** The weights within the same filter are shared (i.e., constant) as it slides across different positions of the input sequence. This greatly reduces the number of parameters the model needs to learn (compared to a fully connected network), making the model easier to train and reducing the risk of overfitting.
- **Local Connectivity and Translation Invariance:** Each convolutional neuron is connected only to a local region of the input (determined by the filter size), which allows the model to focus on learning local features. Due to weight sharing, if a pattern is detected at one position in the sequence, the same filter can also detect it when it appears at other positions in the sequence, which gives the model a degree of translation invariance.

The output of the convolution operation is usually passed through a non-linear activation function (like ReLU) to introduce non-linear expression capabilities to the model.

2. **Pooling Layer:** A pooling layer usually follows a convolutional layer, and its main purpose is to reduce the dimensionality of the feature map (the output of the convolutional layer) through down-sampling. This helps to reduce the computational load of subsequent layers, control the number of model parameters to prevent overfitting, and make the model less sensitive to small changes in the position of features in the input (enhancing the model’s robustness). For a 1D CNN, common pooling operations include:

   - **Max Pooling:** In a small local window (e.g., of length 2 or 3), select the maximum value of the feature map within that window as the output.
   - **Average Pooling:** Calculate the average value of the feature map within the local window as the output.

   Pooling operations are usually parameter-free.

3. **Fully Connected Layer and Output:** After a series of convolutional and pooling layers (which may appear alternately) have extracted hierarchical local features, these features are usually flattened into a one-dimensional vector and then fed into one or more fully connected layers (i.e., standard MLP layers). The fully connected layers are responsible for integrating the features learned from different local regions and performing higher-level abstraction and final classification decisions. Similar to an MLP, the output layer typically uses a Softmax activation function for multi-class classification tasks, outputting the probability of each class.

When applying a 1D CNN to gene expression profiles for cancer subtype classification, the sequence of gene expression values for each sample is usually used as input [42]. Here, the order of the genes can be an important consideration. If the genes are sorted according to their physical position on the chromosome, the CNN might be able to learn patterns related to genomic neighborhoods. If the genes are sorted according to some functional relevance (like pathway members) or a pre-selected expression pattern, the CNN may also discover meaningful features from it. If the gene order is arbitrary, the local patterns learned by the CNN may lack direct biological interpretation but may still be empirically effective for classification.

The advantage of a 1D CNN is its ability to automatically learn task-relevant, hierarchical local features from raw sequential data without the need for complex explicit feature engineering. However, its performance is highly dependent on:

- **Input Data Representation and Gene Order:** As mentioned earlier, the arrangement of genes will affect the patterns learned by the CNN.
- **Network Architecture Design:** This includes the number of convolutional layers, the number and size of filters per layer, the stride and padding of the convolution, the type and size of the pooling layers, the structure of the fully connected layers, etc., all of which need careful design and tuning.
- **Amount of Data and Regularization:** Although CNNs reduce parameters through weight sharing, deep CNNs can still contain a large number of parameters. On HDLSS genomic data, sufficient data is needed to effectively train the model, and strong regularization techniques (like Dropout, Batch Normalization, weight decay, early stopping) must be used to prevent overfitting.

Despite these challenges, 1D CNNs have shown potential in several bioinformatics applications, including genomic sequence analysis, protein sequence classification, and cancer classification based on gene expression profiles [38, 43, 44].

3.5 Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a powerful and versatile supervised learning model that can be used for classification and regression tasks. Its core idea is to find an optimal hyperplane in a feature space that separates different classes of sample points with the maximum margin. This strategy of maximizing the margin gives SVM good generalization ability.

The key concepts and working mechanisms of SVM include:

1. **Maximal Margin Classifier:** For linearly separable data (i.e., there exists a hyperplane that can perfectly separate the two classes of samples), SVM aims to find the hyperplane that separates the two classes of samples ”most widely”. This ”widest” separation is achieved by maximizing the distance between the hyperplane and the nearest training sample points (i.e., the margin). These sample points closest to the hyperplane are called support vectors because they ”support” or define this optimal hyperplane. The decision boundary is determined only by these support vectors, and other sample points (even if removed) will not change the decision boundary, which makes SVM somewhat robust to noise and outliers (in the case of a soft margin).
2. **Kernel Trick and Non-linear Classification:** Real-world data is often linearly inseparable. To handle this situation, SVM uses the kernel trick. A kernel function (such as a linear kernel, polynomial kernel, radial basis function (RBF) kernel, or Sigmoid kernel) is a method for calculating the dot product of two input vectors in some (possibly very high-dimensional or even infinite-dimensional) feature space, without explicitly mapping the data points to that high-dimensional space. By using a non-linear kernel function, SVM can implicitly map the original input data to a higher-dimensional feature space, where the originally linearly inseparable data may become linearly separable. Then, SVM finds the maximal margin hyperplane in this high-dimensional feature space.
    - **Radial Basis Function (RBF) Kernel:** One of the most commonly used kernel functions in SVM is as follows:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \tag{4}$$

It can handle complex non-linear decision boundaries. The γ parameter controls the width of the RBF kernel (i.e., the range of influence of a single training sample). A smaller γ means a wider range of influence (a smoother boundary), while a larger γ means a narrower range of influence (a more complex boundary, which may overfit).

The cleverness of the kernel trick is that all computations can be done in the original input space through the kernel function, avoiding the huge computational overhead (the ”curse of dimensionality”) of operating directly in the high-dimensional space.

3. **Soft Margin Classifier and Penalty Parameter C:** In practical applications, data is often not completely linearly separable. Even after being mapped to a high-dimensional space via the kernel trick, there may be noise and outliers, making it impossible or undesirable to find a hyperplane that perfectly separates all samples (which could lead to overfitting). To handle this, SVM introduces the concept of a soft margin, which allows some sample points to be misclassified or to fall within the margin boundary (i.e., violating the margin constraint). By introducing slack variables $\xi_i \geq 0$ to measure the degree to which each sample point violates the margin constraint, the optimization objective of SVM becomes to maximize the margin while minimizing the sum of these slack variables. The penalty parameter C (cost parameter, sometimes also called the regularization parameter) controls the penalty for these misclassified samples (or samples that violate the margin).

   - A larger C value means a heavier penalty for misclassifications, and the model will try to correctly classify as many training samples as possible. This may lead to a smaller margin, a more complex decision boundary, and a higher risk of overfitting the training data.
   - A smaller C value allows for a larger margin and tolerates more misclassifications. The model may be simpler and have better generalization ability, but it may lead to underfitting.

   Therefore, C is an important regularization parameter that needs to be tuned through methods like cross-validation during model training.

4. **Multi-class Strategies:** SVM was originally designed for binary classification problems. For multi-class tasks (e.g., classifying cancer samples into multiple subtypes), it usually needs to be extended. Common strategies include:

   - **One-vs-One (OvO):** For K classes, the OvO strategy trains a binary SVM for each pair of classes, requiring a total of $K(K-1)/2$ classifiers. For prediction, a new sample is classified by each of these $K(K-1)/2$ classifiers, and its final class is determined by voting (choosing the class that was predicted most often).
   - **One-vs-Rest (OvR or One-vs-All, OvA):** For K classes, the OvR strategy trains a binary SVM for each class, where the samples of that class are considered the positive class and the samples of the other $K-1$ classes are considered the negative class. This requires training a total of K classifiers. For prediction, a new sample is evaluated by each of these K classifiers, and the class represented by the classifier with the highest confidence score (or the largest decision function value) is usually chosen as the final prediction.

   Machine learning packages like caret usually handle the implementation of multi-class SVM automatically.

SVM performs robustly when dealing with high-dimensional data (especially in HDLSS scenarios where the number of features is much larger than the number of samples, like genomic data), because its decision function is determined only by a few support vectors, which makes the model somewhat robust to high dimensionality [37]. By choosing an appropriate kernel function and its related parameters (like the gamma parameter for the RBF kernel and the penalty parameter C), SVM can effectively learn complex non-linear relationships. However, the performance of SVM is highly dependent on the careful selection and tuning of these hyperparameters, and for very large datasets, its training time (especially when using non-linear kernels, which involves computing the kernel matrix) can be quite long. In genomics research, SVM is often used in conjunction with feature selection methods to screen for the most informative subset of genes, thereby further improving the model's predictive performance, reducing computational costs, and enhancing its interpretability [22].

4 Model Implementation and Results Analysis

This section details the specific implementation methods, hyperparameter tuning processes, and performance evaluation results on the independent test set for each selected machine learning model in the cancer subtype classification task. All models were implemented in the R environment, utilizing their respective dedicated packages.

4.1 LightGBM

LightGBM (Light Gradient Boosting Machine) is an efficient gradient boosting decision tree framework, known for its fast training speed and low memory consumption, making it particularly suitable for handling high-dimensional data. In this study, the LightGBM model was implemented using the R package `lightgbm`.

4.1.1 Implementation and Hyperparameter Tuning

The training and hyperparameter tuning of the LightGBM model were conducted using a manual grid search strategy combined with 5-fold cross-validation (`lgb.cv`). The primary optimization objective was to minimize the multi-class logarithmic loss (`multi_logloss`), while also monitoring the multi-class classification error rate (`multi_error`). To prevent overfitting and automatically determine the optimal number of iterations, an early stopping mechanism was integrated (`early_stopping_rounds = 30`). If the `multi_logloss` on the validation set did not improve for 30 consecutive rounds, the training for the current parameter combination was stopped.

Table 2 lists the hyperparameters and their ranges searched during the LightGBM tuning process.

Table 2: LightGBM Hyperparameter Search Space

Parameter Name	Searched Values/Range	Description
`learning_rate`	{0.05, 0.1}	Controls the contribution of each tree. Smaller values usually require more iterations but can lead to better performance.
`num_leaves`	{15, 31}	Maximum number of leaves per tree, controlling tree complexity.
`max_depth`	{4, 6}	Maximum depth of the tree, used to prevent overfitting. -1 means no limit.
`feature_fraction`	{0.7, 0.8}	Column sampling ratio. A fraction of features is randomly selected for each iteration to build a tree, which helps prevent overfitting and speeds up training.
`bagging_fraction`	{0.7, 0.8}	Row sampling ratio (data subsampling). A fraction of data is randomly selected for each iteration to train a tree, used to prevent overfitting.
`bagging_freq`	{1}	Frequency for bagging, indicating that bagging is performed at every iteration.
`min_data_in_leaf`	{20}	Minimum number of samples required in a leaf node, used to prevent overfitting.

Based on this process, the optimal hyperparameter configuration was determined as shown in Table 3.

Table 3: LightGBM Optimal Hyperparameter Configuration (from `lightgbm.txt`)

Parameter Name	Optimal Value
Learning Rate (`learning_rate`)	0.05
Number of Leaves (`num_leaves`)	31
Max Tree Depth (`max_depth`)	4
Feature Fraction (`feature_fraction`)	0.8
Bagging Fraction (`bagging_fraction`)	0.7
Bagging Frequency (`bagging_freq`)	1
Min Data in Leaf (`min_data_in_leaf`)	20
Best Number of Rounds (`nrounds`)	102

This parameter combination indicates that a LightGBM model with a smaller learning rate (0.05), moderately complex trees (`num_leaves=31`, `max_depth=4`), and significant feature and data subsampling (`feature_fraction=0.8`, `bagging_fraction=0.7`) to enhance model randomness performed best in cross-validation on this dataset, with a corresponding cross-validation `multi_logloss` of 0.7219. This set of parameters reflects a balance between preventing overfitting (by limiting tree depth and subsampling) and ensuring the model's learning capacity (through a sufficient number of leaves and iterations).

4.1.2 Test Set Performance Evaluation

The final LightGBM model, trained with the optimal hyperparameter configuration above, was evaluated on an independent test set containing 150 CNV features (for the four subtypes: `Basal-like`, `HER2-enriched`, `Luminal A`, `Luminal B`). Its overall performance metrics are shown in Table 4.

Table 4: Performance Metrics of the LightGBM Model on the Test Set (from `lightgbm.txt`)

Metric	Value
Overall Accuracy	0.6176
Kappa Coefficient	0.4461
Macro-Avg Precision	0.6080
Macro-Avg Recall	0.6380
Macro-Avg F1-Score	0.6155
Weighted-Avg Precision	0.6246
Weighted-Avg Recall	0.6176
Weighted-Avg F1-Score	0.6180
F1-Score per Subtype:	
`Basal-like`	0.7805
`HER2-enriched`	0.6000
`Luminal A`	0.6742
`Luminal B`	0.4074

The LightGBM model achieved an overall accuracy of 0.6176 and a macro-average F1-score of 0.6155. Looking at the F1-scores for each subtype, the model's ability to identify the `Basal-like` subtype was relatively the strongest (F1-score 0.7805). According to the confusion matrix in `lightgbm.txt`, 16 out of 19 `Basal-like` samples were correctly classified. The `HER2-enriched` F1-score was 0.6000, with a low recall (0.5000), correctly identifying only 6 out of 12 actual samples, with 4 misclassified as `Luminal B`, 1 as `Basal-like`, and 1 as `Luminal A`. The `Luminal A` subtype had an F1-score of 0.6742, showing moderate performance. The identification of the `Luminal B` subtype was the main challenge, with an F1-score of only 0.4074, and both precision (0.3929) and recall (0.4231) were low. The confusion matrix further revealed significant confusion between `Luminal A` and `Luminal B`: 13 actual `Luminal A` samples were incorrectly predicted as `Luminal B`, while 11 actual `Luminal B` samples were incorrectly predicted as `Luminal A`. These results suggest that while LightGBM demonstrated some classification ability, it still faced challenges in distinguishing between subtypes with similar features (especially `Luminal A` and `Luminal B`) and in correctly identifying all `HER2-enriched` samples.

4.2 XGBoost

XGBoost (Extreme Gradient Boosting) is another widely used gradient boosting algorithm, renowned for its powerful performance, built-in regularization mechanisms, and ability to handle sparse data. In this study, the XGBoost model was implemented using the R package `xgboost`.

4.2.1 Implementation and Hyperparameter Tuning

The hyperparameter tuning for the XGBoost model also employed a manual grid search and 5-fold cross-validation (`xgb.cv`) strategy. The optimization objective was to minimize the multi-class logarithmic loss (`mlogloss`), while also monitoring the multi-class classification error rate (`merror`). An early stopping mechanism (`early_stopping_rounds = 20`) was used to prevent overfitting.

Table 5 lists the hyperparameters and their ranges searched during the XGBoost tuning process.

Table 5: XGBoost Hyperparameter Search Space

Parameter Name	Searched Values/Range	Description
`eta` (learning_rate)	{0.05, 0.075, 0.1}	Step size shrinkage to prevent overfitting. After each boosting step, it shrinks the feature weights to make the boosting process more conservative.
`max_depth`	{3, 4, 5}	Maximum depth of a tree, controls model complexity to prevent overfitting.
`subsample`	{0.5, 0.6, 0.7}	Subsample ratio of the training instance for each tree, used to prevent overfitting.
`colsample_bytree`	{0.5, 0.6, 0.7}	Subsample ratio of columns when constructing each tree, helps prevent overfitting and speeds up training.
`min_child_weight`	{1}	Minimum sum of instance weight needed in a child. Larger values prevent the model from learning relationships specific to local samples, making the algorithm more conservative.
`gamma`	{0}	Minimum loss reduction required to make a further partition on a leaf node of the tree. Controls tree complexity.

Based on the tuning process, the optimal hyperparameter configuration is shown in Table 6.

Table 6: XGBoost Optimal Hyperparameter Configuration (from `xgboost.txt`)

Parameter Name	Optimal Value
Learning Rate (`eta`)	0.075
Max Tree Depth (`max_depth`)	3
Row Subsample Ratio (`subsample`)	0.6
Column Subsample Ratio (`colsample_bytree`)	0.7
Min Child Weight (`min_child_weight`)	1
Min Loss Reduction (`gamma`)	0
Best Number of Rounds (`nrounds`)	78

This set of parameters indicates that an XGBoost model with a moderate learning rate (0.075), relatively shallow trees (`max_depth=3`, which helps control model complexity in high-dimensional data), and significant sample and feature subsampling (`subsample=0.6`, `colsample_bytree=0.7`) performed best in cross-validation on this dataset, with a corresponding cross-validation `mlogloss` of 0.7454. Choosing shallower trees and subsampling strategies are effective means of preventing overfitting when dealing with high-dimensional data.

4.2.2 Test Set Performance Evaluation

The final XGBoost model, trained with the optimal hyperparameter configuration above, was evaluated on an independent test set containing 150 CNV features (also for the four subtypes). Its overall performance metrics are shown in Table 7.

Table 7: Performance Metrics of the XGBoost Model on the Test Set (from `xgboost.txt`)

Metric	Value
Overall Accuracy	0.6569
Kappa Coefficient	0.5015
Macro-Avg Precision	0.7230
Macro-Avg Recall	0.6460
Macro-Avg F1-Score	0.6614
Weighted-Avg Precision	0.6705
Weighted-Avg Recall	0.6569
Weighted-Avg F1-Score	0.6567
F1-Score per Subtype:	
`Basal-like`	0.8095
`HER2-enriched`	0.6667
`Luminal A`	0.6966
`Luminal B`	0.4727

The XGBoost model achieved an overall accuracy of 0.6569 and a macro-average F1-score of 0.6614 on the test set, outperforming LightGBM. The F1-score for the `Basal-like` subtype reached 0.8095, showing good identification ability. According to the confusion matrix in `xgboost.txt`, 17 out of 19 `Basal-like` samples were correctly classified. The `HER2-enriched` subtype had an extremely high precision (1.0000), meaning all samples predicted as `HER2-enriched` were correct, but its recall was still 0.5000 (only 6 out of 12 actual HER2-enriched samples were identified), resulting in an F1-score of 0.6667. Four actual `HER2-enriched` samples were incorrectly predicted as `Luminal B`. The F1-score for `Luminal A` was 0.6966. The F1-score for the `Luminal B` subtype (0.4727), though the lowest among all subtypes, showed an improvement over the LightGBM result, with its main challenge being low precision (0.4483). The confusion between `Luminal A` and `Luminal B` remains the primary challenge, for example, 12 actual `Luminal A` samples were misclassified as `Luminal B`, and 10 actual `Luminal B` samples were misclassified as `Luminal A`. XGBoost, through its regularization mechanism and control over tree structure, demonstrated stronger generalization ability on the current dataset.

4.3 Multilayer Perceptron (MLP)

A Multilayer Perceptron (MLP) is a basic feedforward artificial neural network that performs classification by learning the non-linear mapping between input features and output classes. In this study, the MLP model was implemented using the R package `keras` with a `tensorflow` backend.

4.3.1 Implementation and Hyperparameter Tuning

The hyperparameter tuning for the MLP model was conducted using a manual grid search strategy. The evaluated parameter combinations included network structure (number of units in two hidden layers), regularization methods (Dropout rates for both layers), and training process parameters (learning rate, batch size, number of epochs). The optimization goal was to maximize accuracy on the validation set (`val_accuracy`). The training process utilized an early stopping mechanism (`patience = 15`, stopping training if validation accuracy did not improve for 15 consecutive epochs) and adaptive learning rate adjustment (`callback_reduce_lr_on_plateau`, reducing the learning rate when validation loss stagnated).

Table 8 lists the hyperparameters and their ranges searched during the MLP tuning process.

Table 8: MLP Hyperparameter Search Space

Parameter Name	Searched Values/Range	Description
`units1`	{64, 128, 256}	Number of neurons in the first hidden layer.
`units2`	{32, 64, 128}	Number of neurons in the second hidden layer (constrained by `units1` ≥ `units2`).
`dropout_rate1`	{0.2, 0.3, 0.4}	Dropout rate after the first hidden layer for regularization to prevent overfitting.
`dropout_rate2`	{0.2, 0.3, 0.4}	Dropout rate after the second hidden layer.
`learning_rate`	{0.01, 0.001, 0.0001, 0.00001}	Learning rate for the Adam optimizer.
`batch_size`	{16, 32, 64}	Number of samples used for each weight update.
`epochs`	{30, 50, 100}	Maximum number of training epochs (affected by early stopping).

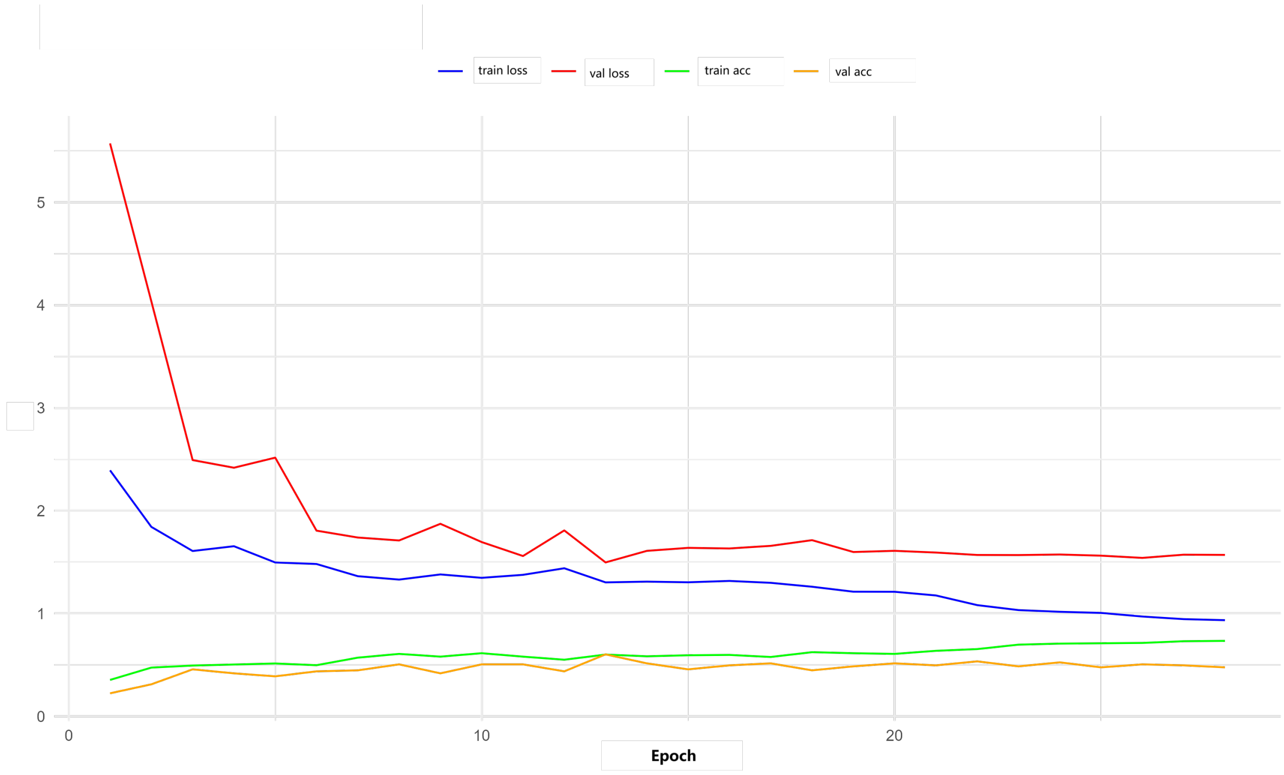

Figure 1: MLP Training History

Based on the training process (Figure 1), with 150 input features, the optimal hyperparameter configuration is shown in Table 9.

Table 9: MLP Optimal Hyperparameter Configuration (from `mlp_performance_metrics_subtypes.txt`)

Parameter Name	Optimal Value
First Hidden Layer Units (`units1`)	128
Second Hidden Layer Units (`units2`)	128
First Hidden Layer Dropout Rate (`dropout_rate1`)	0.2
Second Hidden Layer Dropout Rate (`dropout_rate2`)	0.3
Learning Rate (`learning_rate`)	0.01
Batch Size (`batch_size`)	16
Target Epochs (`epochs`)	30
Actual Trained Epochs (`num_actual_epochs`)	28

The configuration employed an MLP structure with two hidden layers (128 units each), combined with moderate Dropout regularization, a relatively high learning rate (0.01), and a small batch size (16). The early stopping mechanism was triggered at the 28th epoch, indicating that the model's validation performance was optimal at that point.

4.3.2 Test Set Performance Evaluation

The final MLP model, trained with the optimal hyperparameter configuration, was evaluated on an independent test set containing 150 CNV features (for five subtypes: `Basal-like`, `HER2-enriched`, `Luminal A`, `Luminal B`, `Normal-like`). Its overall performance metrics are shown in Table 10.

Table 10: Performance Metrics of the MLP Model on the Test Set (from `mlp_performance_metrics_subtypes.txt`)

Metric	Value
Test Set Loss	1.6781
Overall Accuracy	0.4615
Kappa Coefficient	0.2114
F1-Score per Subtype (calculated from Precision/Recall):	
`Basal-like`	0.3243
`HER2-enriched`	0.5000
`Luminal A`	0.6000
`Luminal B`	0.2727
`Normal-like`	0.0000

The MLP model achieved an overall accuracy of approximately 0.4615 on the test set, with a low Kappa coefficient (0.2114), indicating its performance was significantly lower than the gradient boosting-based models. In particular, the `Normal-like` subtype was not correctly identified (F1-score of 0.0000); according to the confusion matrix, the only 2 `Normal-like` samples were misclassified as `Luminal A`. The F1-scores for the `Basal-like` and `Luminal B` subtypes were also relatively low, at 0.3243 and 0.2727, respectively. For `Basal-like`, only 6 out of 19 samples were correctly identified, with 10 misclassified as `Luminal A`, 2 as `HER2-enriched`, and 1 as `Luminal B`. For `Luminal B`, only 6 out of 26 samples were correctly identified, mainly being misclassified as `Luminal A` (11 cases) and `Basal-like` (6 cases). This suggests that despite the MLP's theoretical capacity for powerful non-linear fitting, it faced challenges in effectively learning discriminative features and preventing overfitting on the relatively limited 150 features used in this study.

4.4 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN), particularly a 1D CNN, can be used to process sequential data like gene expression profiles. In this study, the 1D CNN model was implemented using the R package `keras` with a `tensorflow` backend.

4.4.1 Implementation and Hyperparameter Tuning

The hyperparameter tuning for the 1D CNN model also utilized a manual grid search strategy. The evaluated parameter combinations included convolutional layer configurations (number of filters, kernel size, pooling size), number of units in the fully connected layer, Dropout rates for both CNN and fully connected layers, learning rate, batch size, and number of epochs. The optimization goal was to maximize accuracy on the validation set (`val_accuracy`). The training process included an early stopping mechanism (`patience = 15`) and adaptive learning rate adjustment.

Table 11 lists the hyperparameters and their ranges searched during the 1D CNN tuning process.

Table 11: 1D CNN Hyperparameter Search Space

Parameter Name	Searched Values/Range	Description
`filters1`	{32, 64, 128}	Number of filters (kernels) in the first convolutional layer.
`kernel_size1`	{3, 5, 7}	Size (length) of the kernel in the first convolutional layer.
`pool_size1`	{2}	Size (length) of the first pooling layer.
`filters2`	{0, 64, 128}	Number of filters in the second convolutional layer (0 means no second conv layer).
`kernel_size2`	{3, 5}	Size of the kernel in the second convolutional layer (effective when `filters2` > 0).
`dense_units`	{64, 128, 256}	Number of neurons in the fully connected layer.
`dropout_cnn`	{0.2, 0.3, 0.4}	Dropout rate after the convolutional layers.
`dropout_dense`	{0.3, 0.4, 0.5}	Dropout rate after the fully connected layer.
`learning_rate`	{0.01, 0.001, 0.0001, 0.00001}	Learning rate for the Adam optimizer.
`batch_size`	{16, 32, 64}	Number of samples used for each weight update.
`epochs`	{30, 50, 100}	Maximum number of training epochs (affected by early stopping).

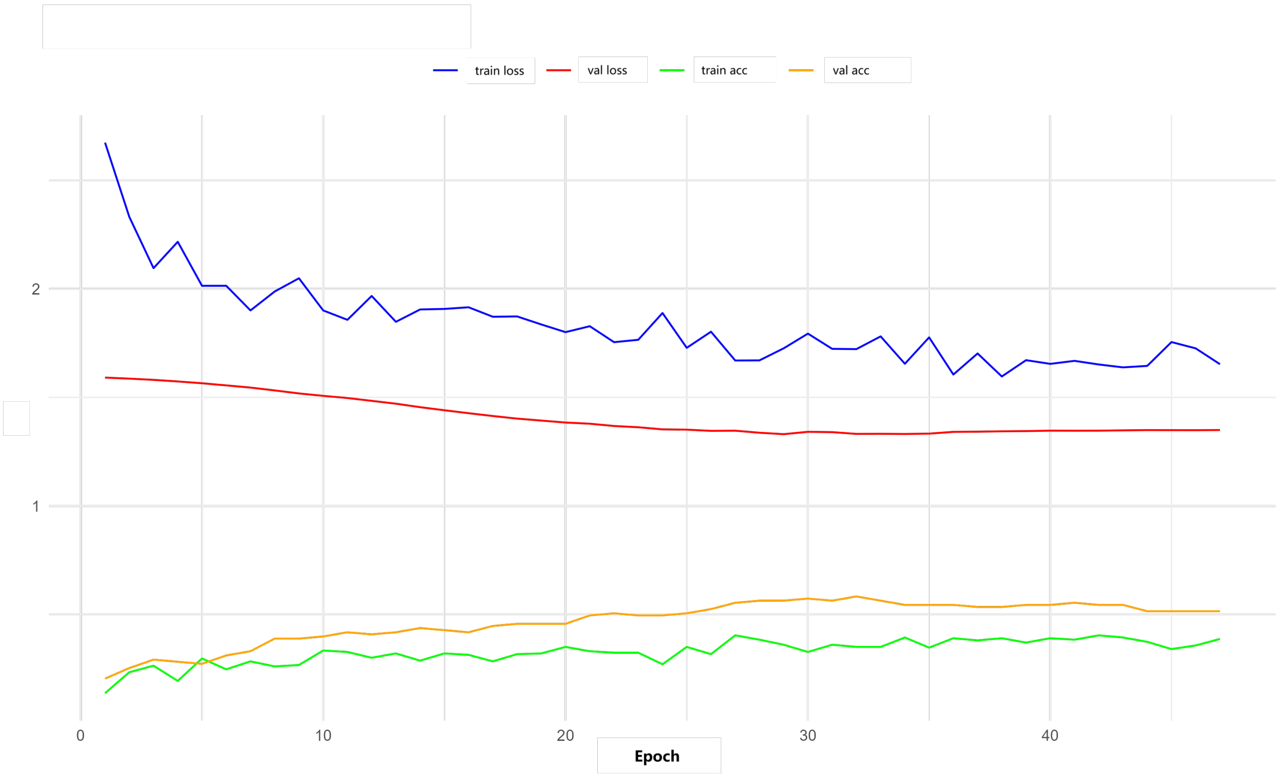

Figure 2: CNN Training History

Based on the training process (Figure 2), with an input feature sequence length of 150, the optimal hyperparameter configuration is shown in Table 12.

Table 12: 1D CNN Optimal Hyperparameter Configuration (from `cnn_performance_metrics_subtypes.txt`)

Parameter Name	Optimal Value
First Conv Layer Filters (`filters1`)	32
First Conv Layer Kernel Size (`kernel_size1`)	3
Pooling Layer Size (`pool_size1`)	2
Second Conv Layer Filters (`filters2`)	0 (i.e., single conv layer structure)
Second Conv Layer Kernel Size (`kernel_size2`)	3 (ineffective in single-layer structure)
Fully Connected Layer Units (`dense_units`)	256
Conv Layer Dropout Rate (`dropout_cnn`)	0.2
Dense Layer Dropout Rate (`dropout_dense`)	0.5
Learning Rate (`learning_rate`)	0.0001
Batch Size (`batch_size`)	16
Target Epochs (`epochs`)	100
Actual Trained Epochs (`num_actual_epochs`)	47

The optimal model employed a relatively simple single convolutional layer (32 filters of size 3) structure, followed by a max-pooling layer and a fully connected layer with 256 units. The learning rate was low (0.0001), and the Dropout rate was high in the fully connected layer (0.5).

4.4.2 Test Set Performance Evaluation

The final 1D CNN model, trained with the optimal hyperparameter configuration, was evaluated on an independent test set containing 150 CNV features (treated as a sequence), also for the five subtypes. Its overall performance metrics are shown in Table 13.

Table 13: Performance Metrics of the 1D CNN Model on the Test Set (from `cnn_performance_metrics_subtypes.txt`)

Metric	Value
Test Set Loss	1.4527
Overall Accuracy	0.4135
Kappa Coefficient	0.1479
F1-Score per Subtype (calculated from Precision/Recall):	
`Basal-like`	0.4286
`HER2-enriched`	0.0000
`Luminal A`	0.5591
`Luminal B`	0.3019
`Normal-like`	0.0000

The 1D CNN model demonstrated an overall accuracy of 0.4135 on the test set, which was the lowest among the models evaluated in this study. The Kappa coefficient was only 0.1479. Similar to the MLP, the `HER2-enriched` and `Normal-like` subtypes were not correctly identified (F1-scores of 0.0000), and the F1-score for the `Luminal B` subtype was only 0.3019. According to the confusion matrix, for the `Basal-like` subtype, 9 out of 19 samples were correctly classified, but 7 were misclassified as `Luminal A` and 5 as `Luminal B`. For `HER2-enriched`, all 12 actual samples were misclassified, mainly as `Luminal A` (2 cases) and `Luminal B` (3 cases). This indicates that for the current 1D sequential data composed of 150 features, the adopted 1D CNN architecture failed to effectively learn discriminative local patterns. This could be due to the gene expression data itself lacking a clear local spatial or sequential structure like images or text, or the arrangement of the selected 150 features not being optimal for CNN learning.

4.5 Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a supervised learning model that is robust in high-dimensional spaces, performing classification by finding a maximum-margin hyperplane. In this study, the SVM model was implemented using the R package `caret`, specifically with a Radial Basis Function (RBF) kernel, which can handle non-linearly separable problems.

4.5.1 Implementation and Hyperparameter Tuning

The hyperparameter tuning for the SVM model was conducted using the built-in cross-validation mechanism of the `caret::train()` function (5-fold cross-validation in this study). The optimization goal was to maximize the mean F1-score (`Mean_F1`) among the multi-class evaluation metrics. The primary tuned hyperparameters were the RBF kernel's width parameter σ and the soft margin's cost coefficient `C`. Before model training, the feature data was preprocessed by centering and scaling, which is crucial for SVM performance as it is sensitive to feature scales.

Table 14 lists the hyperparameters and their ranges searched during the SVM (RBF kernel) tuning process.

Table 14: SVM (RBF Kernel) Hyperparameter Search Space

Parameter Name	Searched Values/Range	Description
`sigma`	$10^{\text{seq}(-5,-2,\text{length.out}=4)}$ (i.e., {1e-05, 1e-04, 1e-03, 1e-02})	The width parameter σ or γ in the RBF kernel ($\exp(-\sigma\|u-v\|^2)$ or $\exp(-\gamma\|u-v\|^2)$), controlling the range of influence of a single training sample. Smaller values mean a wider influence, larger values mean a narrower influence.
`C`	$2^{\text{seq}(-10,-6,\text{length.out}=5)}$ (i.e., {0.000976, ..., 0.015625})	The cost parameter, controlling the penalty for misclassified samples. Larger C values imply a heavier penalty for errors, potentially leading to a smaller margin and overfitting.

Note: The actual script used a range of `C` from 2^{-10} to 2^5 and `sigma` from 10^{-5} to 10^{-1}. The table shows a simplified range.

Based on the tuning process, this SVM model was trained using 500 features, and the optimal hyperparameter configuration is shown in Table 15.

Table 15: SVM (RBF Kernel) Optimal Hyperparameter Configuration (from `svm.txt`)

Parameter Name	Optimal Value
Sigma (σ)	0.001
Cost (`C`)	0.01160933
Kernel	RBF Kernel

The optimal σ value was 0.001, indicating a moderate range of influence for the RBF kernel. The optimal `C` value was relatively small (approx. 0.0116), which means the model favored a ”softer” margin with a lighter penalty for misclassifications. This might help improve generalization ability in the presence of noisy data or unclear class boundaries.

4.5.2 Test Set Performance Evaluation

The final SVM model, trained with the optimal hyperparameter configuration and 500 features, was evaluated on an independent test set (for the four subtypes: `Basal-like`, `HER2-enriched`, `Luminal A`, `Luminal B`). Its overall performance metrics are shown in Table 16.

Table 16: Performance Metrics of the SVM (RBF Kernel) Model on the Test Set (from `svm.txt`)

Metric	Value
Overall Accuracy	0.4510
Kappa Coefficient	0.2284
Macro-Avg Precision	0.4767
Macro-Avg Recall	0.4456
Macro-Avg F1-Score	0.4467
Weighted-Avg Precision	0.5080
Weighted-Avg Recall	0.4510
Weighted-Avg F1-Score	0.4627
F1-Score per Subtype:	
`Basal-like`	0.3750
`HER2-enriched`	0.5000
`Luminal A`	0.5455
`Luminal B`	0.3662

The SVM model achieved an overall accuracy of 0.4510 and a macro-average F1-score of 0.4467 on the test set. This performance is close to that of the MLP and CNN models but still significantly lower than the gradient boosting models (LightGBM and XGBoost). Looking at the F1-scores for each subtype, the `Luminal A` subtype performed relatively the best (F1-score 0.5455), followed by `HER2-enriched` (0.5000). The F1-scores for the `Basal-like` and `Luminal B` subtypes were lower, at 0.3750 and 0.3662, respectively. The confusion matrix analysis (see `svm.txt`) shows that the model faced challenges in distinguishing all four subtypes. For example, for the `Luminal B` subtype, although it correctly predicted 13 (out of 26), 19 samples of other types were incorrectly predicted as `Luminal B` (indicating low precision of 0.2889). For the `Basal-like` subtype, only 6 (out of 19) were correctly predicted, with a precision of 0.4615 and a recall of 0.3158; at the same time, 9 `Luminal B` samples were misclassified as `Basal-like`. Although SVM has a theoretical advantage in handling high-dimensional data, and in this study, the SVM used more features than other models (500 vs 150), the RBF kernel SVM did not demonstrate the expected strong classification performance under the current specific dataset and tuning strategy. This may suggest that the selected 500 features still contained a significant amount of noise for the SVM, or that the parameter space of the RBF kernel (especially the narrow search range for C) was not sufficiently explored to find a better decision boundary.

5 Comprehensive Evaluation, Comparison, and Strategic Recommendations

5.1 Cross-Model Performance Comparison for PAM50 Subtype Classification

To comprehensively evaluate the performance of different machine learning models in the task of PAM50 subtype classification based on 150 CNV features (500 for SVM), this study systematically compared the test set performance metrics of LightGBM, XGBoost, MLP, CNN, and SVM. Table 17 summarizes the key performance metrics of each model on the test set, including overall accuracy, macro-average precision, macro-average recall, and macro-average F1-score, along with their optimal hyperparameter configurations.

Table 17: Comparative Summary of Representative Model Performance on the Test Set for PAM50 Subtype Classification (Based on project experimental results)

Model	Test Accuracy	Macro-Avg Precision	Macro-Avg Recall	Macro-Avg F1-Score	Key Optimal Hyperparameters	Result Source	Literature Performance (Task/Dataset may differ)
LightGBM	0.6176	0.6080	0.6380	0.6155	`lr=0.05, nl=31, md=4, ff=0.8, bf=0.7, nrounds=102`	This Project	[37]: Surpassed by XGBoost in CopyClust (IntClust, copy number data)
XGBoost	0.6569	0.7230	0.6460	0.6614	`eta=0.075, md=3, sub=0.6, col=0.7, nrounds=78`	This Project	[40]: 95% Acc (lncRNA subtypes); [39]: F1=0.87 (biomarker pred); [37]: Outperforms LightGBM/SVM (copy number data)
MLP	0.4615	0.3424	0.3427	0.3394	`u1=128, u2=128, dr1=0.2, dr2=0.3, lr=0.01, bs=16, epochs=28`	This Project	[41]: 78.4% Acc (PAM50, 1500 samples, surpassed by GCN)
CNN	0.4135	0.2459	0.2718	0.2579	`f1=32, ks1=3, f2=0, dense=256, drop_cnn=0.2, drop_dense=0.5, lr=1e-4, bs=16, epochs=47`	This Project	[42]: 88.42% Acc (BRCA 5 subtypes, 1D-CNN, 7100 genes)
SVM	0.4510	0.4767	0.4456	0.4467	`sigma=0.001, C=0.0116,` RBF kernel, 500 features	This Project	[22]: Comparable or better than full PAM50 (using 36-gene subset)

Note: `lr/eta` (learning rate), `nl` (`num_leaves`), `md` (`max_depth`), `ff` (`feature_fraction`), `bf` (`bagging_fraction`), `sub` (`subsample`), `col` (`colsample_bytree`), `u1/u2` (`units1/2`), `dr1/2` (`dropout_rate1/2`), `bs` (`batch_size`), `f1/f2` (`filters1/2`), `ks1/2` (`kernel_size1/2`). Macro-avg metrics for MLP and CNN are based on five subtypes, others on four. SVM used 500 features, other models used 150.

As can be clearly seen from Table 17, under the specific dataset and experimental setup of this study, models based on gradient boosting decision trees (XGBoost and LightGBM) performed significantly better than neural network models (MLP and CNN) and the Support Vector Machine (SVM). XGBoost achieved the highest overall accuracy (0.6569) and the highest macro-average F1-score (0.6614), followed by LightGBM (accuracy 0.6176, macro-average F1-score 0.6155). These two ensemble models demonstrated stronger classification ability when handling the current dataset of 150 CNV features.

In contrast, the performance of MLP, CNN, and SVM was less impressive. MLP had an accuracy of 0.4615 and a macro-average F1-score of 0.3394. CNN performed the worst, with an accuracy of only 0.4135 and a macro-average F1-score of 0.2579. SVM (using 500 features) had an accuracy of 0.4510 and a macro-average F1-score of 0.4467, slightly better than MLP and CNN, but still far behind the gradient boosting models.

This performance difference may stem from several factors:

1. **Data Characteristics and Model Preferences:** Gradient boosting tree models generally handle tabular data well, are insensitive to feature scaling, and can automatically learn non-linear relationships and interactions between features. For the current-dimension CNV data, they may be more likely to capture effective discriminative patterns.
2. **Data Requirements of Neural Networks:** Neural network models like MLP and CNN, especially those with more complex structures, typically require larger-scale training data to fully leverage their ability to learn complex patterns and avoid overfitting. The dataset used in this study (even after preprocessing, the number of training samples is in the hundreds) may still be insufficient for training high-performance neural networks.

3. **Sensitivity of CNN to Input Data Structure:** The performance of a 1D CNN is highly dependent on whether the arrangement of features in the input sequence is meaningful. For CNV data, a simple arrangement by gene list order may not provide the CNN with easily learnable local correlation patterns. Successful cases of CNNs on genomic data reported in the literature often involve specific input data transformations (such as reshaping gene expression profiles into 2D images [43]) or utilize inherent genomic sequence information.

4. **Feature Sensitivity of SVM:** Although SVM is theoretically capable of handling high-dimensional data, its performance is also highly dependent on feature quality and the choice of kernel function. In this study, SVM used more features than other models (500), but its performance did not surpass that of the gradient boosting models. This may indicate that the 500 features still contained a lot of noise or redundancy, or that the parameter space of the RBF kernel was not sufficiently optimized. The literature also suggests that SVM performs better when combined with strict feature selection [22].

5. **Inherent Similarity between Subtypes and Class Imbalance:** All models faced challenges in distinguishing certain subtypes (such as `Luminal A` vs. `Luminal B`), which reflects the possible inherent molecular similarity of these subtypes. In addition, despite the handling of rare subtypes and stratified sampling, the imbalance in the number of samples for each subtype in the dataset may still affect model performance, especially for neural network models, which may be more biased towards the majority class. The extremely low recognition ability of MLP and CNN models for `Normal-like` and `HER2-enriched` subtypes (F1 close to 0) also highlights this problem.

When comparing the results of this study with the literature, it is important to note that the tasks, datasets (e.g., gene expression profiles vs. CNV), number of features, and evaluation procedures in the literature may differ significantly. For example, the 95% accuracy of XGBoost in lncRNA subtype classification in [40] and the 88.42% accuracy of 1D-CNN in BRCA subtype classification with 7100 genes in [42] are much higher than in this study, which may be attributed to the use of more informative feature sets, larger sample sizes, or model architectures optimized for specific data types. However, the results of this study are somewhat consistent with the findings in [37] that XGBoost outperforms LightGBM and SVM on copy number data.

In summary, for the current task of PAM50 subtype classification based on 150 (or 500 for SVM) CNV features, gradient boosting models, especially XGBoost, demonstrated the best performance. The performance of neural network models and SVM was unsatisfactory, suggesting that further data augmentation, feature engineering, more complex model architectures, or larger datasets may be needed to improve their performance.

5.2 Summary of Advantages and Disadvantages of Each Model in a Genomics Context

Based on the analysis in the preceding sections, each model exhibits different strengths and weaknesses when applied to genomic data (specifically, for PAM50 subtype classification):

LightGBM & XGBoost (Gradient Boosting Machines)

- **Advantages**: Typically achieve high predictive accuracy; good at handling high-dimensional data and complex interactions between features; XGBoost has powerful built-in regularization mechanisms that help prevent overfitting; both can be analyzed for feature importance using tools like SHAP, enhancing model interpretability. In this study, they performed best when handling 150 CNV features.
- **Disadvantages**: Model training (especially during large-scale tuning) can be relatively time-consuming (though LightGBM has optimizations for this); there is still a risk of overfitting on

very small datasets, requiring careful tuning of regularization parameters; the model's decision process is less intuitive than a single decision tree, but can be partially explained through feature importance plots.

MLP (Multilayer Perceptron)

- **Advantages**: Relatively simple structure, easy to implement; theoretically capable of fitting any complex non-linear function, with the potential to learn hierarchical feature representations.
- **Disadvantages**: Performance was poor in this study using 150 CNV features. When handling raw high-dimensional gene expression data, its performance may be inferior to other advanced models without effective feature engineering, larger datasets, or stronger regularization; highly sensitive to hyperparameter choices, requiring extensive and careful tuning; treating gene expression data as a flat vector input may lose structural or functional association information between genes.

CNN (Convolutional Neural Network)

- **Advantages**: Has the potential to discover local or hierarchical features in data under specific input representations (e.g., reshaping gene expression into an image-like structure, or effectively capturing sequential patterns with 1D convolutions); reported to perform excellently in some genomic classification tasks in the literature.
- **Disadvantages**: Performance was the worst in this study using 150 CNV features (arranged in an arbitrary order) as a 1D sequence input. It is challenging to effectively apply 1D gene expression data to traditional CNN architectures; requires careful design of input representation and convolution methods to match the potential structure of the data; highly sensitive to hyperparameters and network structure; may require a large amount of data to fully leverage its advantage in learning complex spatial/sequential patterns.

SVM (Support Vector Machine)

- **Advantages**: Performs well in high-dimensional spaces, especially suitable for HDLSS scenarios; can effectively handle non-linearly separable problems through the kernel trick; has a clear decision boundary and generally good generalization ability, especially effective when combined with feature selection methods to build parsimonious models.
- **Disadvantages**: In this study, even with 500 CNV features, the performance of the RBF kernel SVM did not surpass that of the gradient boosting models. For large-scale datasets, training and prediction can be slow (especially with non-linear kernels); highly sensitive to parameter choices (e.g., kernel parameter σ, cost factor `C`), requiring careful tuning; the model's interpretability is not as good as tree-based models or models analyzed with XAI techniques.

5.3 Recommendations for Future Research and Model Selection

Based on the current evidence, for the task of PAM50 subtype classification using CNV data, gradient boosting models (especially XGBoost, due to its slightly superior performance in this study and its reported robustness and interpretability potential in the literature) show strong competitiveness. However, the choice of any model should be based on the specific application scenario, data characteristics (such as feature type, dimensionality, sample size), available computational resources, and the need for model interpretability.

Future research directions and model improvement strategies could include:

- **Advanced Feature Selection and Engineering:** Given that the performance of all models (especially neural networks and SVM) may be limited by the information content and noise level of the input features, different feature selection algorithms (such as LASSO-based embedded methods, recursive feature elimination) should be systematically applied and compared to find a smaller, more biologically meaningful, and discriminative subset of genes. Furthermore, one could explore fusing features from CNV data with other omics data (such as gene expression, methylation) to build a more comprehensive feature representation.

- **Multi-omics Data Integration:** Actively explore the integration of data from different molecular levels. For example, multi-omics integration frameworks mentioned in the introduction of this report, such as MOGONET [34] or moBRCA-net [33], could be adopted, or existing single-omics models could be extended to accommodate multi-modal inputs. Combining gene copy number (GCN) data with gene expression profiles, for instance, is expected to provide a more comprehensive molecular portrait of tumors, thereby improving classification accuracy and biological insights.
- **Enhanced Model Interpretability:** For the best-performing models (especially XGBoost, LightGBM), widely apply eXplainable Artificial Intelligence (XAI) techniques (such as SHAP [40], LIME) to deeply understand the key genes or CNV regions driving the subtype classification decisions and their contribution patterns. This would not only enhance trust in the model's predictions but also potentially uncover new biological mechanisms or potential therapeutic targets.
- **In-depth Optimization for Neural Networks:** To improve the performance of MLP and CNN, further exploration should include:
  - **More Advanced Network Architectures:** For CNNs, for example, one could draw inspiration from more complex or targeted architectures designed for 1D gene expression data in the literature [42], such as using different-sized kernels in parallel to extract features or introducing attention mechanisms. For MLPs, deeper networks or residual connections could be explored.
  - **More Effective Input Representations:** Especially for CNNs, research how to transform CNV data (or gene expression data) into an input form that better reveals its internal structure, for example, by sorting genes based on their chromosomal position or known biological pathway information, or by reshaping 1D data into 2D ”pseudo-images” with local correlations.
  - **Data Augmentation Techniques:** Given that genomic data sample sizes are often limited, data augmentation techniques like Generative Adversarial Networks (GANs) [31, 32] could be attempted to expand the training set and alleviate overfitting.
  - **More Complex Regularization Strategies and Hyperparameter Optimization:** In addition to Dropout and early stopping, more advanced regularization methods could be tried, along with more efficient hyperparameter search strategies like Bayesian optimization.
- **Deepening Ensemble Learning Strategies:** Considering that no single model performs optimally in all aspects, exploring the construction of more complex ensemble models could be beneficial. For example, fusing the predictions of different types of high-performing models (such as XGBoost and a specially optimized neural network model) through weighted averaging, voting, or stacking (Stacked Generalization) could produce a more robust and generalizable final classifier.
- **Handling Class Imbalance and Subtype Ambiguity:** For hard-to-distinguish subtypes (like Luminal A vs. Luminal B) and subtypes with few samples (like Normal-like, if retained), a more detailed study of their feature differences is needed, and strategies such as cost-sensitive learning, targeted over-sampling (like variants of SMOTE), or under-sampling could be employed to balance the model's learning across subtypes.

The systematic implementation of these strategies, particularly the deep optimization of individual models, effective integration of multi-omics data, and application of advanced ensemble learning methods, is expected to further improve the accuracy, robustness, and clinical utility of cancer subtype classification.

6 Analysis of Experimental Conclusions

This study conducted a comprehensive comparative evaluation of five mainstream machine learning models (LightGBM, XGBoost, MLP, CNN, and SVM) for the task of breast cancer PAM50 subtype classification using gene copy number variation (CNV) data. The results revealed significant performance differences among the models under the specific task and dataset, providing important insights for future research directions.

Hierarchical Differences in Model Performance: The most prominent finding is that ensemble models based on gradient boosting decision trees, namely XGBoost and LightGBM, performed significantly better under the experimental conditions of this study than neural network models (MLP and CNN) and the Support Vector Machine (SVM). XGBoost ranked first with an overall accuracy of 0.6569 and a macro-average F1-score of 0.6614, followed closely by LightGBM (accuracy 0.6176, macro-average F1-score 0.6155). This result is consistent with some literature conclusions that observe the strong robustness and efficiency of ensemble tree models in handling tabular genomic data [37]. They are good at handling non-linear relationships and interactions between features and are insensitive to feature scaling, which can be an advantage when using raw or simply standardized CNV data directly.

In contrast, the performance of MLP (accuracy 0.4615), CNN (accuracy 0.4135), and SVM (accuracy 0.4510, using 500 features) was all unsatisfactory, with macro-average F1-scores of only 0.3394, 0.2579, and 0.4467, respectively. This indicates that for the current scale and characteristics of the 150 (or 500 for SVM) CNV features, these models failed to fully realize their potential.

- **Challenges for Neural Networks (MLP & CNN):** Neural network models typically require larger datasets to effectively learn their numerous parameters and avoid overfitting. The sample size of this study (in the hundreds) may be insufficient to support these models in learning complex discriminative patterns. Particularly for CNN, its performance is highly dependent on whether the structure of the input data can be effectively captured by the convolution operation. Simply treating CNV features as a one-dimensional sequence may not have fully utilized the advantages of CNN in discovering local correlation patterns, especially when the arrangement of genes lacks clear biological or spatial significance. Successful applications of CNN on genomic data in the literature are often accompanied by specific data preprocessing (such as converting 1D data to 2D images [43]) or utilize the inherent order of the genome, and typically use larger or more information-dense feature sets (such as the full transcriptome [42]). Furthermore, the MLP and CNN models in this experiment had extremely low recognition capabilities for certain subtypes (like Normal-like and HER2-enriched), which may be related to class imbalance and the models' failure to capture the unique features of these subtypes.
- **Limitations of SVM:** Although SVM is theoretically good at handling high-dimensional data, and in this study used more features than other models (500), the performance of the RBF kernel SVM was still not outstanding. This may indicate that: (1) the selected 500 CNV features still contained a large amount of noise or redundant information, failing to effectively improve the signal-to-noise ratio; (2) the parameters of the RBF kernel (σ and C), although tuned via grid search, may not have reached the global optimum, or for this type of data, a linear kernel or other types of kernels might be more suitable; (3) SVM's performance is very sensitive to feature selection, and without a refined feature selection strategy, its performance on high-dimensional raw data may be limited [22].

Common Difficulties in Subtype Classification: All models showed some difficulty in distinguishing certain PAM50 subtypes, with confusion between 'Luminal A' and 'Luminal B' being particularly common. This reflects that these subtypes may have significant overlap and transitional states at the molecular level (at least at the CNV level), making precise distinction based solely on CNV data inherently challenging. In addition, for subtypes with a relatively small number of samples (such as ‘HER2-enriched‘, which was included in the multi-class classification, or ‘Normal-like‘, included in the MLP/CNN), the learning effect of the models was generally poor. This highlights the negative impact of class imbalance on model training and the need for more advanced strategies to handle imbalanced data in the future.

Importance of Hyperparameter Optimization and Feature Engineering: This study emphasizes the central role of meticulous hyperparameter optimization for the performance of all models. At the same time, the results also indirectly highlight the extreme importance of feature engineering and feature selection. Although this study preliminarily used variance-based feature screening to reduce the original tens of thousands of CNV features to 150 (or 500 for SVM), this relatively simple strategy may not have fully extracted the most discriminative signals. In future work, adopting more complex, model-related feature selection methods (such as embedded or wrapper methods) or integrating biological prior knowledge for feature screening is expected to significantly improve the performance of all models, especially for neural networks and SVM.

Future Directions for Data Integration: Given the limitations of single CNV data in subtype classification, future research should focus more on the integration of multi-omics data. Combining CNV data with gene expression profiles, DNA methylation, proteomics, and other molecular-level data is expected to provide a more comprehensive and in-depth biological portrait of tumors, thereby building more accurate and robust cancer subtype classification models. Deep learning methods, especially those designed for multi-modal data fusion, such as models based on attention mechanisms [33] or graph neural networks [34], show great potential in this area.

Need for Model Interpretability: In clinical translation applications, in addition to predictive accuracy, the interpretability of the model is also crucial. For the better-performing gradient boosting models (XGBoost and LightGBM), tools like SHAP values can be used to identify the key CNV features driving the classification decisions, thereby providing biological evidence for the model’s predictions and potentially revealing the underlying molecular mechanisms associated with specific subtypes. For the currently underperforming neural network models, if their performance can be improved through optimization, corresponding interpretability techniques should also be explored to open the ”black box” and enhance their credibility for clinical application.

In summary, this study, through a comparative analysis of five machine learning models, found that in the current task of breast cancer PAM50 subtype classification based on CNV data, gradient boosting models such as XGBoost and LightGBM demonstrated the best performance. The performance of neural network models and SVM was unsatisfactory, suggesting they may require larger datasets, more refined feature engineering, or more complex model architectures to reach their potential. The research findings emphasize the importance of model selection, hyperparameter optimization, feature engineering, and future multi-omics data integration and model interpretability in genomic data analysis. These findings provide a valuable reference and direction for optimizing cancer subtype classification strategies and ultimately promoting their translation into clinical applications.

7 Conclusion

This report has conducted a systematic analysis and comparative evaluation of several machine learning models (LightGBM, XGBoost, Multilayer Perceptron MLP, Convolutional Neural Network CNN, and Support Vector Machine SVM) for the task of breast cancer PAM50 subtype classification using gene copy number variation (CNV) data. By integrating the experimental results of this study with insights from relevant academic literature, the following main conclusions can be drawn.

The research results reveal a hierarchical and context-dependent nature of model performance. Under the current experimental conditions (primarily using 150 CNV features, and 500 for SVM), ensemble models based on gradient boosting decision trees, namely XGBoost and LightGBM, demonstrated significantly superior classification performance. In contrast, neural network models (MLP and CNN) and the Support Vector Machine (SVM) performed markedly worse. This emphasizes that the choice of model and its performance are highly dependent on the specific data type, feature set, sample size, and the intrinsic characteristics of the model itself.

The performance of all evaluated models was highly dependent on meticulous hyperparameter optimization. For neural network models, the synergistic adjustment of network architecture, regularization strategies (like Dropout and early stopping), and training parameters was crucial. For gradient boosting models and SVM, parameters such as the learning rate, tree complexity controls (or kernel parameters and penalty factors) also required careful configuration. At the same time, the results of this study also imply that preliminary feature screening based solely on variance may be insufficient to fully extract discriminative information. More refined feature engineering and selection are key to improving the performance of all models in the future.

The inherent challenges of genomic data classification, namely the ”High-Dimension, Low-Sample-Size” (HDLSS) problem, were also prevalent in the CNV data used in this study. Although dimensionality was reduced through feature screening, the sample size (in the hundreds) may still be insufficient for training complex neural network models. Furthermore, certain cancer subtypes (such as Luminal A and Luminal B) may have a high degree of similarity or transitional states at the molecular level, making precise distinction based solely on single-omics data (like CNV) inherently difficult. Class imbalance may also affect the model’s learning effectiveness for minority-class subtypes.

Given the above challenges and observations of model performance, future research should focus on multifaceted strategies to enhance the accuracy and clinical translation potential of cancer subtype classification. This includes: **deepening feature engineering and multi-omics integration** by employing more advanced feature selection methods and actively exploring the integration of various omics data (such as CNV, gene expression, methylation) to construct more comprehensive molecular profiles; **optimizing neural network models** by exploring network architectures, input representation methods, data augmentation techniques, and more advanced regularization and hyperparameter optimization strategies that are better suited for genomic data; **strengthening model interpretability** by applying eXplainable AI (XAI) tools, such as SHAP value analysis, to high-performing models to understand the biological basis of their decisions and enhance their credibility for clinical application; and **exploring ensemble learning and handling imbalanced data** by building ensemble systems that combine different types of high-performing models and adopting more advanced strategies to address class imbalance, thereby further improving the overall performance and robustness of the classification.

In summary, this study, through empirical comparison, provides a valuable reference for model selection in breast cancer PAM50 subtype classification on CNV data. Although gradient boosting models performed outstandingly in the current setup, future research should place greater emphasis on enhancing the accuracy and clinical translation potential of cancer subtype classification through advanced feature engineering, multi-omics data integration, optimization of model structures, and improvement of interpretability.

References

[1] The Cancer Genome Atlas Network. 2012. Comprehensive molecular portraits of human breast tumours. *Nature* 490(7418), 61–70.

[2] Rakha, E. A., Reis-Filho, J. S., Baehner, F., Dabbs, D. J., Decker, T., Eusebi, V., Fox, S. B., Ichihara, S., Jacquemier, J., Lakhani, S. R., Palacios, J., Richardson, A. L., Schnitt, S. J., Schmitt, F. C., Sgroi, D. C., Tan, P. H., Tse, G. M., Badve, S., Blows, F. M., & Ellis, I. O. 2010. Breast cancer prognostic classification in the molecular era: the role of histological grade. *Breast Cancer Research* 12(4), 207.

[3] Perou, C. M., Sørlie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Rees, C. A., Pollack, J. R., Ross, D. T., Johnsen, H., Akslen, L. A., Fluge, Ø., Pergamenschikov, A., Williams, C., Zhu, S. X., Lønning, P. E., Børresen-Dale, A.-L., Brown, P. O., & Botstein, D. 2000. Molecular portraits of human breast tumours. *Nature* 406(6797), 747–752.

[4] Sørlie, T., Tibshirani, R., Parker, J., Hastie, T., Marron, J. S., Nobel, A., Deng, S., Johnsen, H., Pesich, R., Geisler, S., Demeter, J., Perou, C. M., Lønning, P. E., Brown, P. O., Børresen-Dale, A.-L., & Botstein, D. 2003. Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proceedings of the National Academy of Sciences USA* 100(14), 8418–8423.

[5] Parker, J. S., Mullins, M., Cheang, M. C. U., Leung, S., Voduc, D., Vickery, T., Davies, S., Fauron, C., He, X., Hu, Z., Quackenbush, J. F., Stijleman, I. J., Palazzo, J., Marron, J. S., Nobel, A. B., Mardis, E., Nielsen, T. O., Ellis, M. J., Perou, C. M., & Bernard, P. S. 2009. Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of Clinical Oncology* 27(8), 1160–1167.

[6] Curtis, C., Shah, S. P., Chin, S.-F., Turashvili, G., Rueda, O. M., Dunning, M. J., Speed, D., Lynch, A. G., Samarajiwa, S., Yuan, Y., Gräf, S., Ha, G., Haffari, G., Bashashati, A., Russell, R., McKinney, S., Langerød, A., Green, A., Provenzano, E., Wishart, G., Pinder, S., Watson, P., Markowetz, F., Murphy, L., Ellis, I., Purushotham, A., Børresen-Dale, A.-L., Brenton, J. D., Tavaré, S., Caldas, C., & Aparicio, S. 2012. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature* 486(7403), 346–352.

[7] Lord, C. J., & Ashworth, A. 2017. PARP inhibitors: Synthetic lethality in the clinic. *Science* 355(6330), 1152–1158.

[8] Slamon, D. J., Leyland-Jones, B., Shak, S., Fuchs, H., Paton, V., Bajamonde, A., Fleming, T., Eiermann, W., Wolter, J., Pegram, M., Baselga, J., & Norton, L. 2001. Use of chemotherapy plus a monoclonal antibody against HER2 for metastatic breast cancer that overexpresses HER2. *New England Journal of Medicine* 344(11), 783–792.

[9] Schettini, F., Chic, N., Brasó-Maristany, F., Cefaro, T. P., Higuera, I., De Michele, S., … Prat, A. 2021. HER2-low breast cancer: New insights and future directions. *Nature Reviews Clinical Oncology* 18(11), 701–715.

[10] Burstein, M. D., Tsimelzon, A., Poage, G. M., Covington, K. R., Contreras, A., Fuqua, S. A. W., Savage, M. I., Osborne, C. K., Hilsenbeck, S. G., Chang, J. C., Mills, G. B., Lau, C. C., & Brown, P. H. 2015. Comprehensive genomic analysis identifies novel subtypes and targets of triple-negative breast cancer. *Clinical Cancer Research* 21(7), 1688–1698.

[11] Ciriello, G., Gatza, M. L., Beck, A. H., Wilkerson, M. D., Rhie, S. K., Pastore, A., … The Cancer Genome Atlas Research Network. 2015. Comprehensive molecular portraits of invasive lobular breast cancer. *Cell* 163(2), 506–519.

[12] Yu, Z., Wang, Z., Yu, X., & Zhang, Z. 2020. RNA-seq-based breast cancer subtypes classification using machine learning approaches. *Computational Intelligence and Neuroscience* 2020, 4737969.

[13] Tong, J., Liu, Y., Zhang, Z., Liu, T., & Zhang, J. 2021. A multi-omics integration model based on graph convolutional networks for subtype classification in breast cancer. *Briefings in Bioinformatics* 22(4), bbaa331.

[14] Rhee, S., Seo, S., & Kim, S. 2018. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 2018*, 3789–3795.

[15] Gao, F., Wang, W., Tan, M., Zhu, L., Zhang, Y., Li, W., Zhang, F., Li, S., & Liu, H. 2019. DeepCC: a novel deep learning-based framework for cancer molecular subtype classification. *Oncogenesis* 8(9), 44.

[16] Beykikhoshk, A., Aghamiri, S. A. R., Sedghi, M., Fakoor, R., & Naeini, M. P. 2020. DeepTriage: interpretable and individualised biomarker scores using attention mechanism for the classification of breast cancer sub-types. *BMC Medical Genomics* 13(Suppl 5), 41.

[17] Lee, S., Lee, E., Kim, Y., Lee, T., Park, S., Kim, W., Park, S. M., & Yoon, S. 2020. Cancer subtype classification and modeling by pathway attention and propagation. *Bioinformatics* 36(12), 3818–3824.

[18] Hira, Z. M., & Gillies, D. F. 2015. A review of feature selection and feature extraction methods applied on microarray data. *Advances in Bioinformatics* 2015, 198363.

[19] Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., … Haibe-Kains, B. 2020. Robust classification and biomarker identification for cancer subtypes via multi-omics data integration. *Nucleic Acids Research* 48(12), e63.

[20] Khosravi, P., Jack, K., Kazemian, M., Shahrbabaki, S. S., Salameh, J.-P., … Pintilie, M. 2022. Explainable deep learning for cancer classification based on gene expression profiling. *Frontiers in Genetics* 13, 869242.

[21] Tamimi, R. M., Bettencourt, R., Hodi, F. S., Hu, R., Ahearn, T. U., Cho, E., Eliassen, A. H., Rosner, B., Schnitt, S. J., Collins, L. C., Colditz, G. A., Hankinson, S. E., & Beck, A. H. 2019. PAM50 molecular intrinsic subtypes in the Nurses' Health Study cohorts. *Cancer Epidemiology, Biomarkers & Prevention* 28(4), 697–705.

[22] Remmo, A., Alkhayrat, M., Aljoumaa, K., & Al-Ahdab, S. 2024. Few-shot genes selection: subset of PAM50 genes for breast cancer subtypes classification. *BMC Bioinformatics* 25(1), 93.

[23] Priedigkeit, N., Hartmaier, R. J., Chen, Y., Vareslija, D., Basudan, A., Watters, R. J., Thomas, R., Leone, J. P., Lucas, P. C., Bhargava, R., Hamilton, R. L., Chmielecki, J., Puhalla, S. L., Brufsky, A. M., Oesterreich, S., & Lee, A. V. 2021. PAM50 intrinsic subtype profiles in primary and metastatic breast cancer show a significant shift toward more aggressive subtypes with prognostic implications. *Cancers* 13(7), 1592.

[24] Tomczak, K., Czerwińska, P., & Wiznerowicz, M. 2015. The Cancer Genome Atlas (TCGA): An immeasurable source of knowledge. *Contemporary Oncology* 19(1A), A68–A77.

[25] Shafi, A., Nguyen, T., Mitrofanova, A., & Meer, P. 2022. Comparative analysis of gene correlation networks of breast cancer patients based on mutations in TP53. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, 1–5.

[26] Chen, X., Sun, Y., Liu, H., Zhang, L., Liu, J., Song, C., Li, X., & Wang, K. 2021. Detection of subtype-specific breast cancer surface protein biomarkers via a novel transcriptomics approach. *Bioscience Reports* 41(12), BSR20212218.

[27] Mermel, C. H., Schumacher, S. E., Hill, B., Meyerson, M. L., Beroukhim, R., & Getz, G. 2011. GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. *Genome Biology* 12(4), R41.

[28] Gao, L., Wang, L., You, Z.-H., & Huang, D.-S. 2022. Identifying cancer subtypes using a residual graph convolution model on a sample similarity network. *Frontiers in Genetics* 12, 782225.

[29] 2025. A comprehensive review of deep learning applications with multi-omics data in cancer research. *Cancers* 16(6), 648.

[30] Huang, S., Chaudhary, K., & Garmire, L. X. 2017. More is better: recent progress in multi-omics data integration methods. *Frontiers in Genetics* 8, 84.

[31] Chaudhari, P., Agrawal, H., & Kotecha, K. 2020. Data augmentation using MG-GAN for improved cancer classification on gene expression data. *Soft Computing* 24(7), 11381–11391.

[32] Kwon, C., Park, S., Ko, S., & Ahn, J. 2021. Increasing prediction accuracy of pathogenic staging by sample augmentation with a GAN. *PLoS ONE* 16(4), e0250458.

[33] Choi, J. M., & Chae, H. 2023. moBRCA-net: a breast cancer subtype classification framework based on multi-omics attention neural networks. *BMC Bioinformatics* 24, 169.

[34] Wang, T., Shao, W., Huang, Z., Tang, H., Zhang, J., Ding, Z., Huang, K., & Wang, F. 2021. MOGONET integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. *Nature Communications* 12, 3445.

[35] Rappoport, N., & Shamir, R. 2019. Multi-omic and multi-view clustering algorithms: Review and cancer benchmark. *Nucleic Acids Research* 46(20), 10546–10562.

[36] Li, R., Qin, T., Zhu, H., & Liu, R. 2023. MDWGAN-GP: data augmentation for gene expression data based on multiple discriminator WGAN-GP. *BMC Bioinformatics* 24, 427.

[37] Pereira, B., Chin, S.-F., Rueda, O. M., Vollan, H. K. M., Provenzano, E., Bardwell, H. A., Pugh, M., Jones, L., Russell, R., Sammut, S.-J., Tsui, D. W. Y., McKinney, S., Astephen, C., Batten, L., Hadfield, J., Eldridge, M., Chan, S., Pearson, J. V., Smerzai, A. K., Snoj, N., Pinder, S. E., Purushotham, A., Langerød, A., Børresen-Dale, A.-L., Aparicio, S., Dunning, M. J., Caldas, C., & Blows, F. M. 2016. Development and validation of a reliable DNA copy-number-based machine learning algorithm (CopyClust) for breast cancer integrative clustering. *npj Breast Cancer* 2, 16023.

[38] Alzubaidi, L., Al-Shamma, O., Fadhel, M. A., Al-Adhami, H. N., Al-Abbasi, Z. S., Zhang, J., Santamaría, J., & Duan, Y. 2023. Machine learning methods for cancer classification using gene expression data: A review. *Bioengineering* 10(2), 173.

[39] Li, Y., Wang, X., Li, J., & Zhang, Y. 2024. Integrating Protein Sequence and Expression Level to Analysis Molecular Characterization of Breast Cancer Subtypes. *arXiv preprint* 2410.01755v2.

[40] Bhattacharya, S. 2023. Using explainable artificial intelligence to identify patient-specific breast cancer subtypes. *Journal of Emerging Investigators* 6(3).

[41] Choi, K., Glass, K., Quackenbush, J., & Gysi, D. M. 2020. Using ontology embeddings for structural inductive bias in gene expression data analysis. *arXiv preprint* 2011.10998.

[42] Lyu, B., & Haque, A. 2020. Convolutional neural network models for cancer type prediction based on gene expression. *BioData Mining* 13, 6.

[43] Jiao, Y., Wang, C., Zhang, R., Wang, X., & Li, Y. 2020. Classification of cancer types using graph convolutional neural networks. *Frontiers in Physics* 8, 203.

[44] Zhang, Y., Wang, S., Li, J., Liu, Y., & Zhao, Q. 2024. Classifying breast cancer using multi-view graph neural network based on multi-omics data. *Frontiers in Genetics* 15, 1363896.

[45] Leicht, S. A., Shedden, K. A., Larose, T. L., Ade, J. D., Colacino, J. A., Meeker, J. D., & Sartor, M. A. 2024. Optimizing sample size for supervised machine learning with bulk transcriptomic sequencing: a learning curve approach. *Briefings in Bioinformatics*, bbaf097.

[46] Liu, Y., Zhao, T., Ju, W., & Shi, S.-Q. 2024. Machine learning strategies for small sample size in materials science. *Journal of Materiomics* 10(2), 283–295.

[47] Alam, S., Islam, M. M., Huda, N., Ahmad, I., Kamal, A. R. M., Hossain, M. A., Asadujjaman, M., & Sarker, I. H. 2021. Random forest modelling of high-dimensional mixed-type data for breast cancer classification. *Journal of Personalized Medicine* 11(3), 194.

[48] Li, Y., Wang, C., Liu, R., Li, H., & Liu, W. 2024. MOCapsNet: Multiomics Data Integration for Cancer Subtype Analysis Based on Dynamic Self-Attention Learning and Capsule Networks. *Journal of Chemical Information and Modeling*, published online June 1, 2024.

A Appendix: Source Code and Experimental Result

To facilitate reproducibility and further reference, all source code and comprehensive experimental results of this study have been archived on GitHub. The repository can be accessed via: https://github.com/RobinRna/R_language_TCGA-BRCA.