People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

KASDI MERBAH UNIVERSITY - OUARGLA

Faculty of New Technologies of Information and Communication

Department of Computer Science and Information Technology

MASTER THESIS

Domain: Mathematics and Computer Science

Field: Computer Science

Speciality: Artificial Intelligence and Data Science

Theme

Fusion CNN and ViT with self-Attention for Multi-Label ECG Classification

---

By: Ad Manel and Boublal Ihssane

Jury Members:

Supervisor: Dr. Aiadi Oussama

President: Dr. Hamrouni Lamis

Examiner: Dr. Korichi Aicha

Academic Year: 2024/2025

Acknowledgment

First and foremost, we thank **Allah the Almighty**, the Greatest of all, for His guidance and facilitation in completing this thesis. We are deeply grateful for His countless blessings, and we hope this work will benefit us in both our religion and our worldly life.
We also express our sincere gratitude to **Dr. Oussama Aiadi**, our respected supervisor, for his continuous support, sage advice, insightful criticism, and encouragement throughout our research journey. His belief in us, valuable feedback, and guidance have been instrumental in shaping this work. We are truly blessed to have had the opportunity to work under his supervision, and we deeply appreciate his confidence in us.
Likewise, we extend our respectful thanks to the esteemed **jury members** for accepting to review our work. We are honored to have you evaluate our research, and we sincerely thank you in advance for your insightful remarks and contributions.
Our heartfelt thanks go to our **parents, families, and siblings** for their unwavering support, unconditional love, patience, and understanding. Their continuous moral support has been the driving force behind the completion of this thesis, and we are forever indebted to them.
We also wish to express our gratitude to all our **teachers, professors, and faculty members** at the Department of Computer Science and Information Technology, Kasdi Merbah University. Your dedication to teaching, your encouragement, and the academic environment you foster have not only enriched our knowledge but have also shaped us into better individuals.
We are thankful for the opportunity to study alongside our esteemed **colleagues and fellow students**. Their cooperation, constructive feedback, and enriching discussions have significantly contributed to the development and refinement of this research.
Before concluding, we extend our gratitude once more to every individual who supported us during this journey. Your encouragement, contributions, and unconditional love mean more to us than words can express.
In the end, may **Allah (SWT)** accept this humble effort as a good deed and grant us all success in this life and the hereafter. *Ameen.*

Abstract

Cardiovascular diseases remain the leading cause of mortality worldwide, underscoring the critical importance of accurate and timely diagnosis. Electrocardiography (ECG) is a widely used, non-invasive technique for detecting cardiac abnormalities, yet its manual interpretation remains challenging—particularly in multi-label scenarios where several co-occurring conditions may be present in a single recording.

This thesis addresses these challenges by investigating a range of machine learning (SVM, Random Forest, XGBoost) and deep learning architectures (CNN, LSTM, Transformer), with a particular focus on multi-label ECG classification using the PTB-XL dataset. We propose a hybrid deep learning model combining Convolutional Neural Networks (CNN), Transformer layers, and a Self-Attention mechanism. This architecture was specifically designed to improve the classification of both common and underrepresented cardiac conditions while maintaining computational efficiency.

Experimental results demonstrate that the proposed hybrid model outperforms both standalone deep learning and classical machine learning models. It achieved a Binary Accuracy of 93.32%, a Micro F1-score of 0.7663, and a recall of 0.7238, while reducing false negatives in classes such as Myocardial Infarction (MI) and Hypertrophy (HYP). In addition, the model integrates Focal Loss to mitigate class imbalance and employs explainability techniques (XAI) to enhance interpretability in clinical applications.

These findings confirm the potential of hybrid architectures in delivering robust, scalable, and interpretable solutions for real-world ECG classification tasks.

Keywords: ECG classification, deep learning, multi-label, CNN, Transformer, Self-Attention, Focal Loss, XAI.

Résumé

Les maladies cardiovasculaires demeurent la première cause de mortalité dans le monde, ce qui souligne l'importance cruciale d'un diagnostic précis et rapide.
L'électrocardiogramme (ECG) est une technique non invasive largement utilisée pour détecter les anomalies cardiaques, mais son interprétation manuelle reste complexe, en particulier dans des scénarios multi-étiquettes où plusieurs pathologies peuvent coexister dans un seul enregistrement.
Ce mémoire aborde ces défis en étudiant plusieurs modèles d'apprentissage automatique (SVM, Random Forest, XGBoost) et d'apprentissage profond (CNN, LSTM, Transformer), appliqués à la classification multi-étiquettes des signaux ECG à partir de la base de données PTB-XL. Nous proposons une architecture hybride combinant des réseaux de neurones convolutifs (CNN), des couches Transformer et un mécanisme d'attention (Self-Attention), afin d'améliorer la détection des affections fréquentes et sous-représentées, tout en assurant une efficacité computationnelle optimale.
Les résultats expérimentaux montrent que le modèle hybride proposé surpasse les modèles classiques et profonds utilisés séparément. Il a atteint une précision binaire de 93,32%, un F1-score micro de 0,7663 et un rappel de 0,7238, tout en réduisant les faux négatifs pour des classes critiques telles que l'infarctus du myocarde (MI) et l'hypertrophie (HYP). En outre, la perte Focal Loss a été utilisée pour compenser le déséquilibre des classes, et des techniques d'explicabilité (XAI) ont été intégrées pour améliorer l'interprétabilité clinique.
Ces travaux démontrent le potentiel des architectures hybrides pour fournir des solutions robustes, interprétables et applicables à grande échelle à la classification ECG dans des contextes cliniques réels.
Mots-clés : Classification ECG, apprentissage profond, multi-étiquettes, CNN, Transformer, Self-Attention, Focal Loss, XAI.

ملخص

تُعد أمراض القلب والأوعية الدموية السبب الرئيسي للوفيات على مستوى العالم، مما يجعل التشخيص المبكر والدقيق ضرورة قصوى. يُعتبر تخطيط القلب الكهربائي (ECG) أداة غير جراحية وفعالة لرصد نشاط القلب، إلا أن تفسيره يدوياً لا يزال تحدياً، خصوصًا في الحالات متعددة التسميات التي تتضمن وجود أكثر من اضطراب قلبي في تسجيل واحد.
يتناول هذا العمل هذه التحديات من خلال دراسة مقارنة لعدة نماذج من التعلم الآلي (SVM، Random Forest، XGBoost) والتعلم العميق (CNN، LSTM، Transformer) باستخدام قاعدة البيانات PTB-XL. كما نقترح نموذجًا هجيناً يدمج بين الشبكات العصبية الالتفافية (CNN) وطبقات Transformer، وآلية الانتباه الذاتي (Sefl-Attention) بهدف تحسين تصنيف كل من الحالات الشائعة والنادرة، مع ضمان كفاءة حسابية عالية.
أظهرت النتائج التجريبية تفوق النموذج الهجين المقترح مقارنةً بالنماذج الأخرى، حيث حقق دقة ثنائية بنسبة 93.32%، ودرجة F1 مصغّرة بلغت 0.7663، واسترجاعًا 0.7238، مع تحسين ملحوظ في اكتشاف الحالات القلبية غير المتوازنة مثل احتشاء عضلة القلب (MI) وتضخم القلب (HYP). كما تم استخدام خوارزمية Loss Focal للتعامل مع مشكلة عدم توازن الفئات، بالإضافة إلى توظيف تقنيات التفسير (XAI) لتعزيز موثوقية النموذج في البيئات السريرية.
تؤكد هذه الدراسة فعالية النماذج الهجينة في تقديم حلول دقيقة، قابلة للتفسير، وقابلة للتطبيق في تصنيف إشارات ECG في الواقع السريري.
الكلمات المفتاحية: تصنيف ECG، التعلم العميق، متعدد التسميات، CNN، Transformer، الانتباه الذاتي، Focal Loss، التفسير (XAI).

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

1.1 Introduction

Cardiovascular diseases (CVDs) remain the leading cause of death globally, accounting for approximately 17.9 million deaths annually—about 32% of all global fatalities [74]. These conditions encompass a broad spectrum of cardiac abnormalities, including myocardial infarction (MI), arrhythmias, conduction disturbances (CD), hypertrophy (HYP), and ischemic heart disease. Timely and accurate diagnosis is essential for improving clinical outcomes and reducing mortality [7].

Electrocardiography (ECG) is a widely used, non-invasive, and cost-effective diagnostic tool for monitoring the heart's electrical activity. Despite its clinical value, manual ECG interpretation is time-consuming, prone to inter-observer variability, and often insufficient when multiple cardiac conditions coexist [23]. These limitations underscore the need for automated, scalable analysis techniques.

Recent advances in artificial intelligence (AI), particularly deep learning (DL), have enabled the development of powerful models capable of automatically extracting complex temporal and morphological patterns from ECG signals. Architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models have demonstrated expert-level performance in various ECG classification tasks [61, 75].

In parallel, classical machine learning (ML) models—such as Support Vector Machines (SVM), Random Forest, and XGBoost—have also shown promise, particularly when used with carefully engineered features and effective preprocessing pipelines [17]. However, their limited capacity to model long-term temporal dependencies may restrict their performance in complex diagnostic scenarios.

Therefore, this thesis investigates and compares several deep learning and traditional machine learning models for multi-label ECG classification. In response to the observed challenges, we propose a novel hybrid architecture combining CNN, Transformer, and Self-Attention components to improve diagnostic accuracy—particularly in handling co-occurring cardiac abnormalities and mitigating class imbalance.

1.2 Problematic

Although electrocardiography (ECG) is a cornerstone of cardiovascular diagnostics, the accurate interpretation of multi-lead ECG signals remains a complex and error-prone

task, especially in the presence of co-occurring cardiac conditions. Manual interpretation is prone to observer variability and cognitive overload, motivating the need for automated systems that can deliver robust and scalable multi-label classification.

However, designing such systems presents several challenges. First, ECG signals often reflect multiple overlapping pathologies—such as Myocardial Infarction (MI), Hypertrophy (HYP), and Conduction Disturbances (CD)—within a single recording. This multi-label nature complicates the classification process, requiring models that can independently and jointly recognize multiple conditions per sample. Second, class imbalance is a significant issue, as certain conditions like HYP and CD are underrepresented in the PTB-XL dataset, leading to biased predictions and high false negative rates for minority classes [69].

Traditional machine learning methods such as Random Forest, XGBoost, and SVM have demonstrated utility, but their reliance on handcrafted features and limited temporal modeling capabilities restricts their generalization in high-dimensional biomedical signals [17]. Deep learning architectures like LSTM struggled in our experiments, achieving a recall as low as 31.7% with high volatility across epochs and significant underperformance on MI and HYP classes [45].

Transformer-based models offered improved recall and stability but introduced significant computational overhead, exceeding 2.1 billion FLOPs per prediction [65]. This complexity impedes their use in real-time or resource-constrained clinical settings. Moreover, many models tend to overfit frequent classes like NORM while neglecting subtle morphological and temporal features associated with rare diseases.

These limitations underscore the need for an optimized, interpretable, and computationally efficient model capable of capturing both local morphological features and long-range temporal dependencies. Our hybrid CNN–Transformer–Self-Attention architecture was developed in response to these challenges, aiming to enhance sensitivity for minority classes and provide balanced performance across all diagnostic categories.

1.3 Overview of the Related Techniques

Numerous techniques have been developed for the classification of electrocardiogram (ECG) signals, ranging from traditional machine learning (ML) methods to advanced deep learning (DL) architectures and hybrid models. Each approach offers distinct advantages and faces unique challenges when applied to multi-lead, multi-label ECG classification tasks.

Traditional machine learning models—such as Support Vector Machines (SVM), Random Forest (RF), and Extreme Gradient Boosting (XGBoost)—have shown promising results, particularly when paired with well-engineered features derived from time-domain, frequency-domain, or wavelet-transformed ECG data [17, 50]. These models are often lightweight and interpretable, making them attractive for clinical settings. However, their reliance on handcrafted features and limited capacity to model complex temporal dependencies can hinder performance, especially in multi-label scenarios.

Deep learning approaches, particularly Convolutional Neural Networks (CNNs), have transformed ECG analysis by automatically extracting hierarchical features from raw signals [34]. CNNs are especially effective at capturing local morphological patterns such as the QRS complex, P and T waves. Long Short-Term Memory (LSTM) networks and other recurrent models, on the other hand, have been employed to model long-term temporal dependencies across signal segments [53]. Despite their theoretical suitability,

our experiments revealed that standalone LSTMs often suffer from unstable training and low sensitivity to minority classes.

Transformer-based models, originally developed for natural language processing, have recently been adapted for ECG signal analysis due to their strength in capturing global dependencies and parallel processing [76]. While Transformers improve performance on complex multi-label tasks, they also introduce high computational costs and can be prone to overfitting on small or imbalanced datasets.

Hybrid architectures attempt to integrate the strengths of both CNNs and Transformers. In this context, several studies have explored CNN–Transformer combinations to balance local feature extraction with global attention modeling [40, 63]. Building on this trend, our work proposes an enhanced hybrid model that combines CNN, Transformer, and Self-Attention mechanisms. This architecture aims to improve diagnostic accuracy by capturing both short-term waveform characteristics and long-range temporal relationships, while mitigating issues of class imbalance and model over-complexity.

The evolution from classical ML to deep and hybrid models reflects the ongoing pursuit of accurate, generalizable, and computationally feasible solutions for automated ECG interpretation.

1.4 Motivation

The motivation behind this work stems from the urgent need to improve the accuracy, efficiency, and clinical applicability of ECG-based diagnostic systems—particularly in multi-label settings where multiple co-occurring cardiac conditions must be detected within a single recording.

Manual ECG interpretation, while clinically accepted, is prone to human error and inter-observer variability, especially when subtle waveform abnormalities overlap. These challenges highlight the need for automated systems that can deliver consistent and scalable analysis across diverse patient populations.

Our experimental results revealed limitations in both traditional and deep learning approaches. Classical models like Random Forest and XGBoost, though efficient and interpretable, struggled with capturing temporal dependencies and required extensive feature engineering. LSTM-based architectures demonstrated unstable training behavior and low recall scores for critical but underrepresented classes such as MI and HYP. Although Transformer-based models improved overall accuracy and recall, they introduced significant computational costs, exceeding 2.1 billion FLOPs per prediction—hindering real-time deployment in clinical environments.

To address these issues, we propose a hybrid architecture that integrates CNNs, Transformers, and Self-Attention mechanisms. This design leverages CNNs for local morphological pattern extraction, Transformers for modeling long-range temporal dependencies, and Self-Attention to enhance focus on diagnostically relevant signal regions. Our objective is to deliver a model that balances diagnostic accuracy, generalization across all condition categories, and computational efficiency—paving the way for practical deployment in real-world healthcare settings.

1.5 Contributions

The main contributions of this thesis in the field of automated ECG classification are summarized as follows:

- A comparative study was conducted on several machine learning models (SVM, Random Forest, XGBoost) and deep learning architectures (CNN, LSTM, Transformer) for multi-label ECG classification using the PTB-XL dataset.
- A novel hybrid architecture was proposed and implemented, integrating Convolutional Neural Networks (CNN), Transformer layers, and a Self-Attention module to enhance diagnostic accuracy and generalization across both frequent and underrepresented cardiac conditions.
- The class imbalance issue was addressed by employing Focal Loss during training, which significantly improved recall for minority classes such as Myocardial Infarction (MI) and Hypertrophy (HYP).
- All models were evaluated and compared using multiple metrics including accuracy, F1-score, recall, and computational cost (FLOPs), highlighting trade-offs between predictive performance and deployment feasibility.
- Explainability techniques based on XAI were incorporated to interpret model predictions and improve clinical trust in the system.

1.6 Thesis Structure

This thesis is organized as follows:

- The **General Introduction** presents the context of the work, highlights the challenges of multi-label ECG classification, and outlines the objectives and contributions of the thesis.
- **Chapter 1** provides the necessary background on ECG signals and classification techniques. It reviews classical machine learning methods (SVM, Random Forest, XGBoost) as well as deep learning models (CNN, LSTM, Transformer), and introduces related concepts such as attention mechanisms, Focal Loss, and explainable AI (XAI).
- **Chapter 2** details the proposed methodology, including several model architectures explored in this work. These include standalone CNN, LSTM, Transformer, as well as hybrid combinations such as CNN + Transformer and CNN + Transformer + Self-Attention. Each model is described with its motivation and design considerations aimed at addressing multi-label ECG classification challenges.
- **Chapter 3** presents the experimental setup and results. It includes preprocessing steps, evaluation metrics, and performance comparisons between the proposed model and baseline ML/DL models, supported by detailed analysis.
- The **Conclusion and Future Work** summarizes the main findings, highlights the contributions of the study, discusses its limitations, and suggests directions for future research.

Chapter 2

Work Background

2.1 Introduction

Electrocardiography (ECG) is a widely used, non-invasive diagnostic technique that records the electrical activity of the heart through surface electrodes placed on the skin. It plays a fundamental role in the detection and monitoring of cardiovascular diseases such as arrhythmias, myocardial infarction, and hypertrophy. Due to its accessibility, affordability, and diagnostic richness, ECG remains a cornerstone in both routine and emergency medical care.

In recent years, the integration of artificial intelligence (AI), particularly machine learning (ML) and deep learning (DL), has significantly enhanced the interpretation of ECG signals. These data-driven approaches enable automated feature extraction, improve diagnostic accuracy, and facilitate early detection of cardiac abnormalities.

This chapter provides the foundational background necessary for understanding the use of AI in ECG classification. It begins with an overview of machine learning paradigms, including supervised, unsupervised, semi-supervised, transfer, and few-shot learning. It then explores traditional ML models such as Support Vector Machines (SVM), Random Forest (RF), and XGBoost, followed by advanced deep learning techniques including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Transformer-based architectures. Attention mechanisms and explainable AI (XAI) tools—such as Grad-CAM, SHAP, and LIME—are also introduced to highlight the importance of interpretability and clinical transparency.

Finally, the chapter concludes with a comparative analysis of recent multi-label ECG classification studies using the PTB-XL dataset, highlighting their methodologies, performance, and limitations, and setting the stage for the development of the proposed approach.

Figure 2.1 illustrates a standard ECG waveform, which captures the heart's electrical activity over time. The waveform is composed of distinct components: the P wave represents atrial depolarization, the QRS complex corresponds to rapid ventricular depolarization, and the T wave reflects ventricular repolarization. These segments occur sequentially and define the cardiac cycle's electrical phases. Accurate identification of these components is essential in diagnosing a range of cardiac conditions, including arrhythmias, myocardial ischemia, and infarction [9, 21].

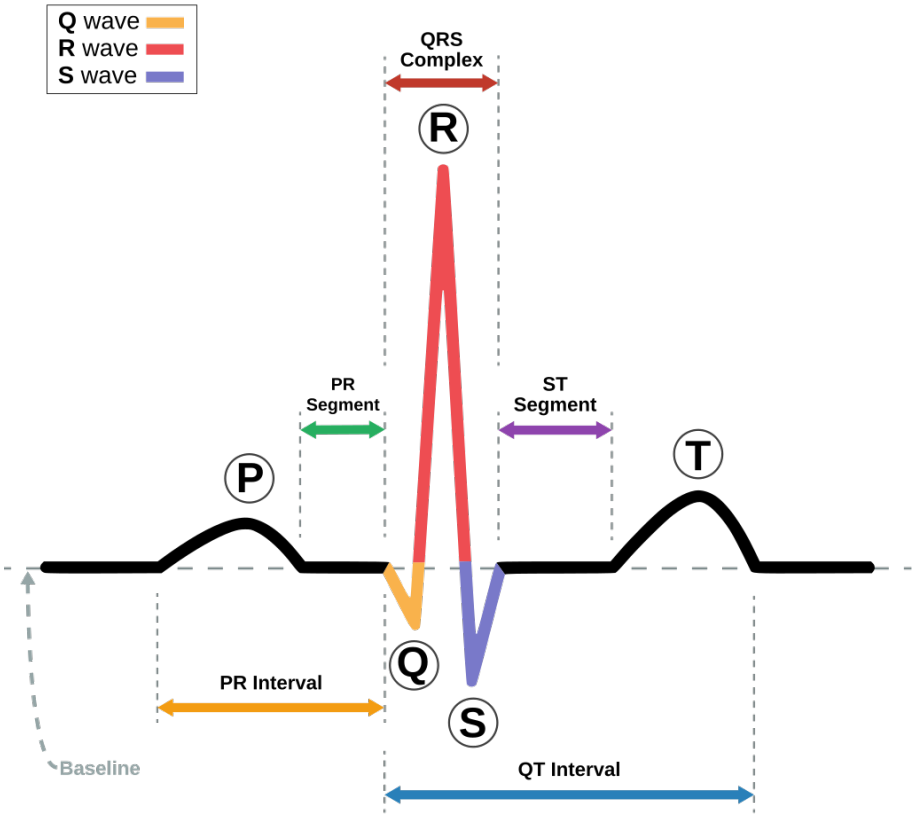

Figure 2.1: Standard ECG waveform illustrating the P wave, QRS complex, and T wave [73].

2.2 Machine Learning

Machine learning (ML) is an umbrella term that refers to a broad range of algorithms that perform intelligent predictions based on a data set. The datasets involved are often vast, potentially comprising millions of unique data points [22]. Machine learning systems are designed to automatically learn and improve from experience without explicit programming [8].

In recent years, the field has witnessed remarkable progress, achieving — and in some cases surpassing — human-level performance in tasks such as semantic understanding, information extraction, and complex pattern recognition [55]. Modern machine learning extends beyond traditional statistical modeling approaches, leveraging the exponential growth in available data, substantial advancements in computational capabilities, and continuous innovations in algorithm development, largely driven by the requirements of web-scale industries [55].

Today, a wide variety of machine learning algorithms, commonly referred to as models, are employed across numerous domains. The choice of an appropriate model is fundamentally influenced by the nature of the data and the specific objectives of the task [8]. For instance, large-scale datasets, consisting of millions of samples, often necessitate the deployment of deep learning architectures [22], whereas smaller datasets are better addressed using classical methods such as linear regression or decision-tree algorithms [8]. Thus, selecting a suitable modeling strategy requires careful consideration of the data type, whether it comprises images, sequential signals, or structured descriptive attributes.

In summary, machine learning constitutes a trans-formative paradigm within artificial intelligence, empowering systems to autonomously derive insights and make data-driven decisions with minimal human intervention [55].

2.2.1 Machine Learning Paradigms

Generally, learning can be supervised, semi-supervised, unsupervised, and reinforcement.

Supervised Learning

In supervised learning, the dataset is the collection of labeled examples $(x_i, y_i)_{i=1}^{N}$. Each element x_i among N is called a feature vector. A feature vector is a vector in which each dimension $j = 1, \ldots, D$ contains a value that describes the example somehow. That value is called a feature and is denoted as $x^{(j)}$.
For instance, if each example x in our collection represents a person, then the first feature, $x^{(1)}$, could contain height in cm, the second feature, $x^{(2)}$, could contain weight in kg, and $x^{(3)}$ could contain gender, and so on. For all examples in the dataset, the feature at position j in the feature vector always contains the same kind of information. It means that if $x_i^{(2)}$ contains weight in kg in some example x_i, then $x_k^{(2)}$ will also contain weight in kg in every example $x_k, k = 1, 2, \ldots, N$.
The label y_i can be either an element belonging to a finite set of classes $1, 2, \ldots, C$, or a real number, or a more complex structure, like a vector, a matrix, a tree, or a graph. Unless otherwise stated, in this book y_i is either one of a finite set of classes or a real number.
You can see a class as a category to which an example belongs. For instance, if your examples are email messages and your problem is spam detection, then you have two classes $\{spam, not\_spam\}$.
The goal of a supervised learning algorithm is to use the dataset to produce a model that takes a feature vector x as input and outputs information that allows deducing the label for this feature vector. [2]
Therefore, to predict whether an email message is spam or not spam using a Support Vector Machine (SVM) model, the process involves several steps. First, the text of the email is converted into a feature vector. This feature vector is then multiplied by w, the weight vector, and the bias term b is subtracted. The sign of the resulting value determines the classification:
If the result is +1, the email is classified as spam.
If the result is -1, the email is classified as not spam.
The next step is determining how the machine finds the optimal values for w and b. This is achieved by solving an optimization problem, as machines are particularly efficient at optimizing functions under constraints.
The primary constraint in this case is ensuring that the model correctly classifies the 10,000 training examples. Each example i, where $i = 1, \ldots, 10,000$, is represented by a pair (x_i, y_i), where x_i is the feature vector and y_i is the corresponding label, taking values either +1 (spam) or -1 (not spam).
The constraints for correct classification are formally defined as follows:

$$w \cdot x_i - b \geq 1, \quad \text{if } y_i = +1$$

$$w \cdot x_i - b \leq -1, \quad \text{if } y_i = -1$$

These constraints ensure that all spam emails (where $y_i = +1$) are positioned on one side of the decision boundary, while all non-spam emails (where $y_i = -1$) are on the other

side, maximizing the margin between the two classes.This process is illustrated in Figure 2.2.

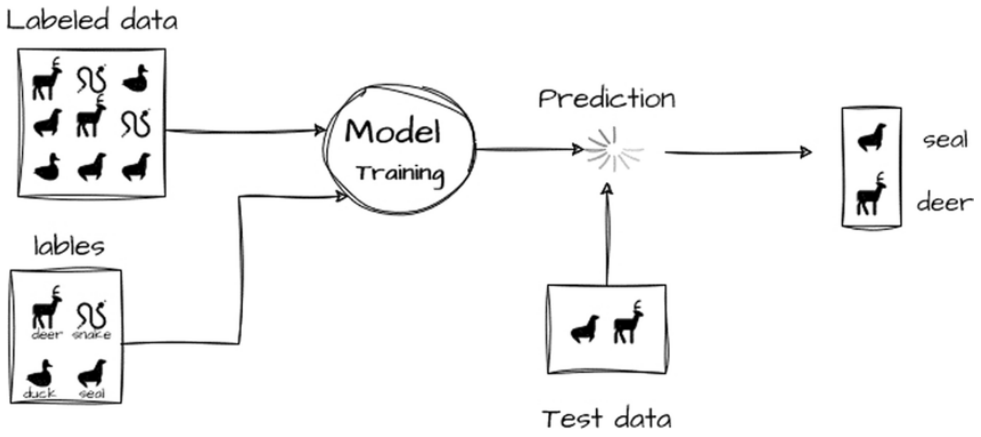

Figure 2.2: Architecture of supervised learning [32]

For example, a commonly used supervised learning model is the **Multilayer Perceptron (MLP)**. It consists of layers of interconnected neurons where each layer applies a linear transformation followed by a non-linear activation. The output of the MLP provides a prediction for the input vector, and the model is trained by minimizing the prediction error on the labeled dataset. The mathematical operation at each layer is defined as:

$$a^{(l)} = \sigma\left(W^{(l)}a^{(l-1)} + b^{(l)}\right)$$

We will examine the Multilayer Perceptron more formally and in greater detail in the subsequent chapter.

Unsupervised Learning

In unsupervised learning, the dataset is a collection of unlabeled examples $\{x_i\}_{i=1}^N$. Unlike supervised learning, where each example has a corresponding label, in unsupervised learning, we only have feature vectors without explicit labels.
The goal of an unsupervised learning algorithm is to find structure in the data. One common task in unsupervised learning is clustering, where we group similar examples together. Another common task is density estimation, where we try to estimate the probability distribution of the data.
For example, in customer segmentation, unsupervised learning algorithms can be used to group customers based on their purchasing behavior without knowing in advance which categories exist This process is depicted in Figure 2.3.

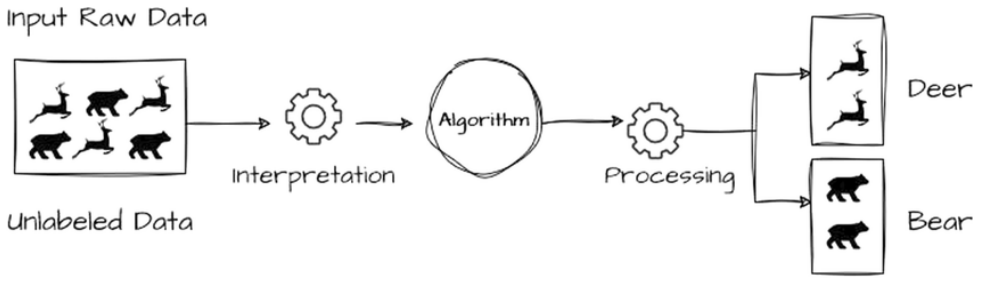

Figure 2.3: Architecture of unsupervised learning [32]

One widely used unsupervised model is K-Means Clustering (Unsupervised Learning)

K-Means is an unsupervised learning algorithm that partitions a dataset into K clusters. Each data point is assigned to the cluster with the nearest mean (centroid). The objective is to minimize the within-cluster sum of squares:

$$\arg\min_{C} \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

Where:

- C_k is the set of points in cluster k,
- μ_k is the centroid of cluster k,
- $\|\cdot\|$ denotes the Euclidean norm.

1.2.3 Semi-Supervised Learning

In semi-supervised learning, the dataset consists of a small number of labeled examples $\{(x_i, y_i)\}_{i=1}^L$ and a large number of unlabeled examples $\{x_j\}_{j=L+1}^N$.
The goal of a semi-supervised learning algorithm is to leverage the large amount of unlabeled data, together with the small labeled dataset, to improve learning performance. This is useful when labeling data is expensive or time-consuming, but obtaining unlabeled data is easy.

For example, in speech recognition, we might have a small set of transcribed audio recordings (labeled data) and a large amount of raw audio without transcriptions (unlabeled data). A semi-supervised learning algorithm can use both to improve accuracy.

Transfer Learning (TL):

Transfer Learning in machine learning is a technique that leverages the knowledge acquired by a model trained on a large dataset in a source task and transfers it to a related but distinct target task, especially when data in the target domain is scarce [51]. Instead of training a model from scratch, TL allows fine-tuning of pre-trained models, reducing both computation time and required labeled data. This approach has been widely adopted in domains such as natural language processing (e.g., BERT, GPT) and computer vision (e.g., ResNet, VGG), where large-scale datasets like ImageNet are used for pretraining. Transfer learning helps improve generalization and model performance when labeled examples in the target task are limited. A schematic representation of the transfer learning workflow can be seen in Figure 2.4.

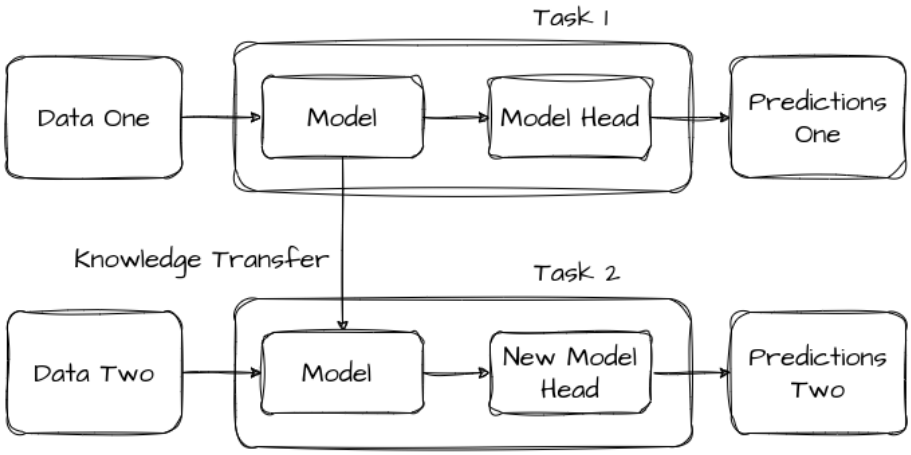

Figure 2.4: Architecture of transfer learning [16]

Few-Shot Learning (FSL):

Few-Shot Learning (FSL) is a subfield of machine learning that aims to train models to generalize effectively from only a small number of labeled examples per class. Unlike traditional methods that require large volumes of data, FSL mimics human-like learning abilities by leveraging prior knowledge from previously learned tasks to make predictions on new, unseen classes with minimal data. This approach is particularly useful in scenarios where collecting and labeling data is expensive, time-consuming, or impractical—such as in medical diagnosis, rare language processing, or wildlife species recognition.

2.3 Machine Learning for ECG Signal Analysis

Machine learning (ML) is a core discipline of artificial intelligence that enables systems to learn patterns and make data-driven decisions without explicit programming. In the context of biomedical signal processing, especially electrocardiogram (ECG) signals, ML

algorithms have shown promising potential in detecting abnormalities, classifying arrhythmias, and predicting disease outcomes [15, 46].
ECG signals are rich in physiological information but often present challenges due to their non-stationary, noisy, and patient-specific nature. Traditional rule-based diagnostic approaches rely heavily on expert-defined features, which may fail to generalize across diverse populations and conditions. Machine learning offers a robust alternative by learning discriminative features directly from the data, thereby reducing reliance on handcrafted features and domain-specific heuristics [3].
Popular ML techniques in ECG analysis include support vector machines (SVM), decision trees, random forests, k-nearest neighbors (KNN), and ensemble learning methods. These models are often trained on features derived from time-domain, frequency-domain, or wavelet-transformed ECG data. While not as data-hungry as deep learning models, classical ML methods still require careful feature selection and preprocessing to perform optimally [36].
In clinical applications, ML-based ECG analysis aids in early detection of conditions like atrial fibrillation, myocardial infarction, and ventricular tachycardia. Moreover, these methods facilitate remote monitoring and automated triage, improving healthcare accessibility and efficiency. As annotated ECG datasets grow in size and diversity, machine learning continues to evolve as a pivotal tool for scalable and reproducible cardiovascular diagnostics [81].

2.3.1 Multi-Layer Perceptrons (MLPs)

Multi-Layer Perceptrons (MLPs) are a class of feedforward neural networks composed of an input layer, one or more hidden layers, and an output layer. Each neuron in these layers performs a weighted sum of its inputs followed by a nonlinear activation function, allowing the network to model complex, non-linear relationships. Mathematically, the output y of a neuron given input vector $\mathbf{x}$, weight vector $\mathbf{w}$, and bias b, is expressed as:

$$y = \phi(\mathbf{w}^T\mathbf{x} + b)$$

where $\phi(\cdot)$ is a non-linear activation function. Two commonly used activation functions are the Rectified Linear Unit (ReLU), defined by:

$$\text{ReLU}(z) = \max(0, z)$$

and the Softmax function, typically used in the output layer for multi-class classification:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

where z_i is the input score (logit) for class i, and K is the number of classes.
In the field of electrocardiogram (ECG) signal analysis, MLPs have been effectively used for tasks such as arrhythmia detection, disease classification, and signal quality assessment. Typically, ECG signals are preprocessed and converted into numerical feature vectors through methods such as statistical analysis, wavelet decomposition, or frequency-domain transformations. These features serve as the input to the MLP, which then learns the mapping between signal characteristics and diagnostic outcomes.
The strengths of MLPs lie in their conceptual simplicity, flexibility, and relatively low computational cost. When trained on well-extracted features, MLPs can achieve high

performance in distinguishing between normal and pathological conditions. Moreover, their efficiency makes them suitable for real-time ECG monitoring and deployment in embedded or resource-constrained systems. This makes MLPs a practical and effective choice in automated ECG interpretation pipelines [12, 20, 28, 78].

2.3.2 Random Forests for ECG Signal Classification

Random Forest (RF) is an ensemble learning method that constructs a collection of decision trees, each trained on a random subset of the training data with replacement (bootstrap sampling). During training, at each node in a tree, a random subset of the input features is selected, and the best split is determined based on an impurity criterion such as *Gini impurity* or *information gain (entropy)*. This randomness both in data and feature selection leads to a diverse set of decision trees whose outputs are aggregated—typically by majority voting for classification tasks [10].
In contrast to neural networks, Random Forests do not rely on *activation functions* like ReLU or Softmax. Instead, each tree produces a *discrete class prediction*, and the ensemble's final decision is based on a *voting mechanism*, which acts as a non-linear function in itself, offering strong generalization capabilities. Additionally, Random Forests do not use a loss function optimized via gradient descent. Instead, they aim to minimize impurity measures locally at each split, making the algorithm more interpretable and less sensitive to hyperparameters compared to deep learning models.
When applied to ECG signal classification, Random Forests rely on engineered features extracted from the ECG waveform, such as time-domain statistics (e.g., RR intervals, P-QRS-T durations), frequency-domain features (via Fourier or wavelet transforms), and morphological descriptors. These features capture essential information about heart rhythm and morphology, which the model learns to associate with pathological or normal patterns.
Random Forests have been used successfully in detecting *arrhythmias*, *ischemic changes*, and even in the early prediction of conditions such as *diabetes* or *sleep apnea* using ECG data [77, 85]. Their ability to rank features by importance also aids clinicians in understanding which aspects of the signal are most indicative of disease, improving transparency in medical decision support systems.
Moreover, their robustness to noise and missing data, combined with low computational cost during inference, makes them well-suited for *real-time ECG monitoring applications* and *embedded health systems*, particularly in resource-constrained environments.

2.3.3 Extreme Gradient Boosting (XGBoost) for ECG Signal Classification

Extreme Gradient Boosting (XGBoost) is an advanced implementation of gradient boosting machines, specifically optimized for efficiency, scalability, and model performance. XGBoost builds an ensemble of weak learners—typically decision trees—by adding them sequentially to minimize a specified loss function. At each boosting iteration, a new tree is trained to predict the residuals (errors) of the ensemble's current predictions.
Unlike Random Forests, which average the outputs of independently trained trees, XGBoost uses a weighted additive model where each tree corrects the errors of the previous ones. The model optimizes a regularized objective function defined as:

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k),$$

where $l(y_i, \hat{y}_i)$ is a convex loss function (e.g., logistic loss for classification), and $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|w\|^2$ is a regularization term that penalizes model complexity, with T being the number of leaves and w the leaf weights [14].

In ECG signal classification, XGBoost is applied after extracting meaningful features from ECG recordings—such as statistical features, heart rate variability metrics, and frequency components derived from signal processing techniques like the Discrete Wavelet Transform (DWT). These features form a high-dimensional input space, and XGBoost excels in selecting and weighting the most informative ones.

Its ability to handle missing data internally, perform automatic feature selection, and model non-linear decision boundaries makes it a robust choice for biomedical signal processing. Several studies have demonstrated the effectiveness of XGBoost in detecting arrhythmias, ischemic episodes, and even predicting non-cardiac conditions such as metabolic syndrome or sleep disorders based on ECG-derived biomarkers [26, 72].

Moreover, XGBoost provides interpretable outputs through feature importance scores and SHAP (SHapley Additive exPlanations) values, which can help domain experts understand the rationale behind model decisions—an essential factor in clinical decision support systems.

2.4 Deep Learning for ECG Signal Analysis

Deep learning (DL) is a subfield of machine learning that employs multi-layered neural networks to automatically learn hierarchical representations from large-scale data. It has demonstrated remarkable performance in complex tasks such as image recognition, speech processing, and biomedical signal analysis [38].

Electrocardiogram (ECG) interpretation is a fundamental task in the diagnosis of cardiovascular diseases, yet it often requires significant expertise due to the complexity and variability of ECG signals. Manual analysis is subject to intra- and inter-observer variability, and subtle morphological changes in the waveforms may go unnoticed, especially in early disease stages.

In recent years, deep learning techniques have emerged as powerful tools for automating ECG analysis. By leveraging large-scale annotated datasets, DL models can learn intricate patterns and temporal dependencies in ECG signals that may not be apparent to human observers. These models—especially convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and Transformer-based architectures—offer robust feature extraction and classification capabilities that enhance diagnostic accuracy, reduce clinician workload, and enable real-time monitoring.

Moreover, deep learning mitigates the need for handcrafted features, which are often limited by domain expertise and subjective interpretation. This shift from rule-based to data-driven approaches is transforming ECG interpretation, offering a promising path toward more objective and scalable diagnostic systems [4, 48, 68].

2.4.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of deep learning models particularly well-suited for processing spatially structured data. Although originally developed for image recognition tasks, CNNs have proven highly effective in one-dimensional signal analysis, such as electrocardiogram (ECG) classification, due to their ability to capture local and hierarchical patterns in sequential data [2, 49].

A typical CNN architecture for ECG classification consists of a stack of one-dimensional convolutional layers (Conv1D), each applying a set of learnable filters (kernels) that slide over the input signal. These filters detect localized features such as peaks, waveforms, and transitions—corresponding to the P-wave, QRS complex, and T-wave [35]. Mathematically, the output of a convolutional layer is computed as:

$$y(t) = (x * w)(t) + b = \sum_{\tau=0}^{k-1} x(t+\tau)w(\tau) + b$$

where $x(t)$ is the input signal, $w(\tau)$ is the filter of size k, and b is the bias term.

Following each convolutional layer, a non-linear activation function such as the Rectified Linear Unit (ReLU) is applied to introduce non-linearity:

$$\text{ReLU}(x) = \max(0, x)$$

This helps the model learn complex, non-linear relationships in the data [38]. To reduce the dimensionality and computation while preserving key features, pooling layers—typically MaxPooling1D—are inserted after activation layers. These layers retain the most prominent feature in each window, increasing robustness to local variations in signal position.

Batch normalization is often applied after convolutions to stabilize and accelerate training by normalizing layer inputs [30]. Additionally, Dropout layers may be used to prevent overfitting by randomly deactivating a fraction of neurons during training [60].

The final part of the CNN architecture includes one or more fully connected (dense) layers, which interpret the learned features for classification. In the context of multi-label ECG classification, a sigmoid activation function is typically used in the output layer to produce independent probabilities for each class:

$$\hat{y}_i = \sigma(z_i) = \frac{1}{1 + e^{-z_i}}$$

The training process minimizes a loss function—commonly binary cross-entropy or focal loss—to optimize the network parameters. Focal loss, in particular, is beneficial when handling class imbalance in ECG datasets, as it down-weights easy examples and focuses training on hard, misclassified cases [42].

An overview of a typical CNN architecture for ECG classification is illustrated in Fig. 3.2, highlighting the key components such as convolutional layers, pooling layers, and fully connected layers.

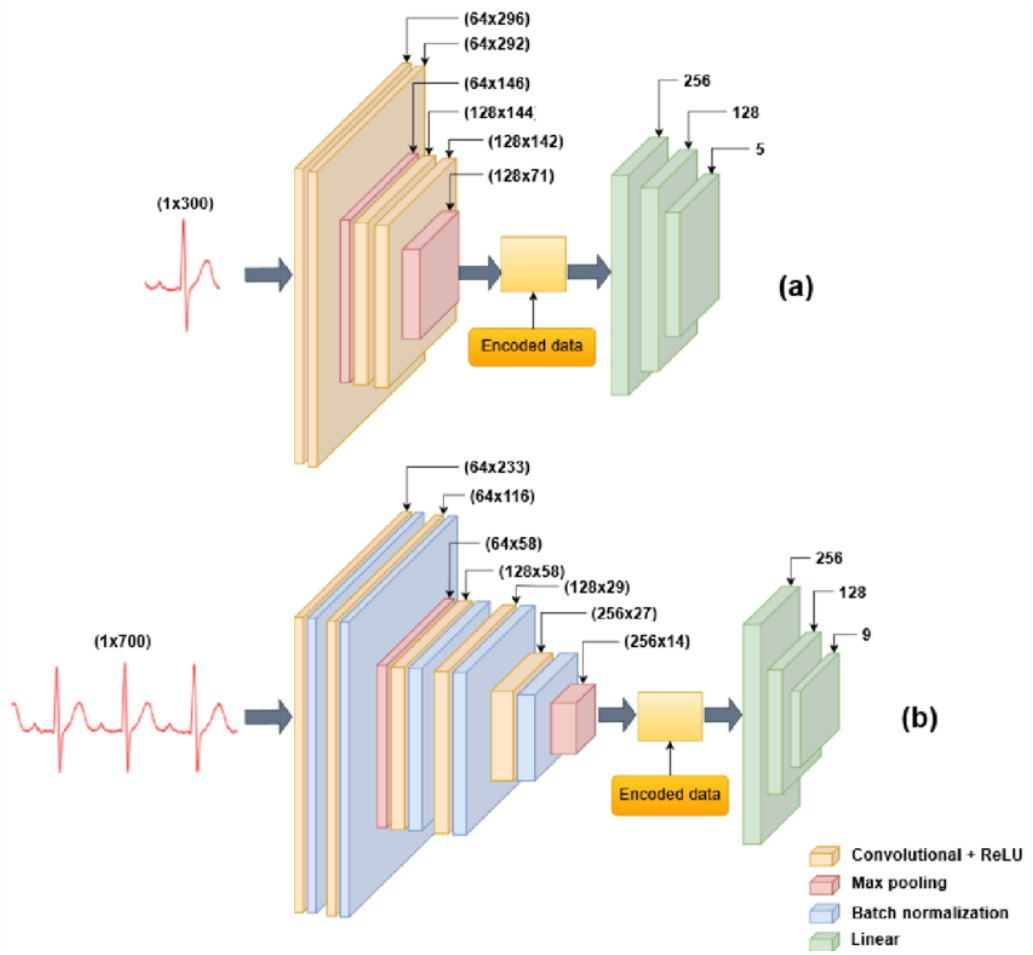

Figure 2.5: A CNN architecture for ECG classification, illustrating convolutional, pooling, and dense layers used to extract features from ECG signals. *Source: adapted from* [49].

2.4.2 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks are a specialized type of recurrent neural networks (RNNs) designed to overcome the vanishing gradient problem in modeling long-range dependencies in sequential data [24]. They have been particularly effective in biomedical time-series tasks, including ECG signal classification, due to their ability to capture the temporal evolution of heartbeats.

An LSTM cell maintains a cell state C_t and a hidden state h_t at each time step t. It uses three gates—*forget gate*, *input gate*, and *output gate*—to control the flow of information. The mathematical formulation is as follows:

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{(forget gate)} \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{(input gate)} \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \text{(candidate state)} \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad \text{(cell state update)} \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(output gate)} \\
h_t &= o_t \odot \tanh(C_t) \quad \text{(hidden state)}
\end{aligned}
$$

Where σ is the sigmoid function, tanh is the hyperbolic tangent, and $\odot$ denotes element-

wise multiplication. These operations allow the network to selectively forget, update, and output information across time steps, enabling robust modeling of temporal dependencies. In the context of ECG classification, LSTMs are used to track the evolution of cardiac signals across multiple time steps. Unlike CNNs, which extract local spatial features, LSTMs provide global temporal context, making them ideal for detecting long-term dependencies such as abnormal rhythms or evolving waveforms. This makes LSTM particularly useful for identifying delayed arrhythmias or recurring wave patterns that span across multiple cardiac cycles.
Figure 2.6 illustrates a typical LSTM-based ECG classification pipeline, showing the stacked LSTM layers followed by dense layers that perform classification based on the extracted temporal features [56].

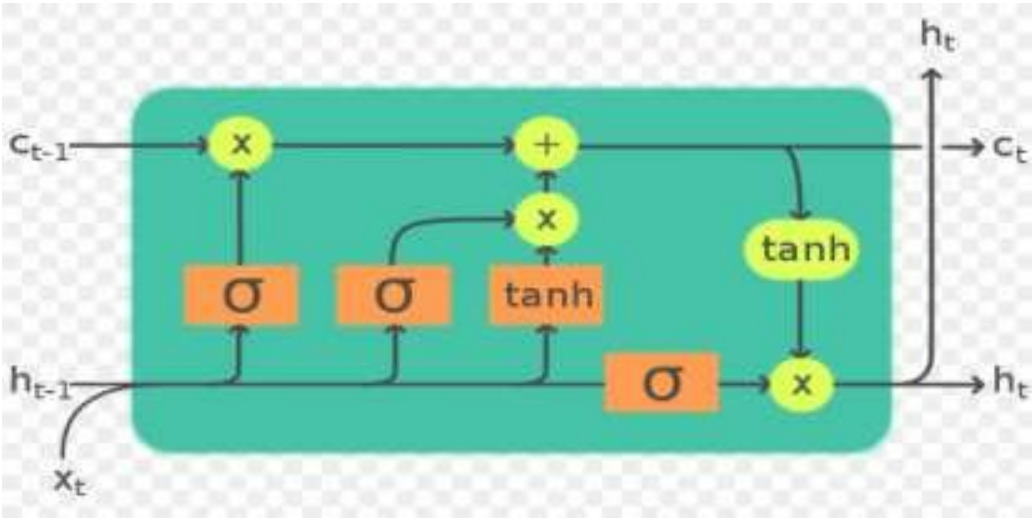

Figure 2.6: An LSTM-based architecture for ECG classification, illustrating the sequence of LSTM layers and fully connected layers used to capture temporal dependencies in ECG signals [56].

2.4.3 Transformer-Based Models

Transformer models, originally developed for natural language processing [65], have recently demonstrated strong performance in biomedical signal analysis, particularly for ECG classification. Unlike CNNs and LSTMs, which rely on local receptive fields or recurrence, Transformers capture long-range dependencies through parallelizable self-attention mechanisms.
In 12-lead ECGs, this global modeling capability helps detect complex temporal and inter-lead dependencies. The attention mechanism enables the model to focus on clinically relevant features such as abnormal QRS complexes or subtle ST-segment deviations [43, 86].
A popular adaptation is the Vision Transformer (ViT), which divides input signals into patches and processes them as sequences. The core components include:

- **Patch Embedding**: Segments the ECG into fixed-length patches and maps each to an embedding vector.
- **Positional Encoding**: Adds sequence order information.

- **Multi-Head Self-Attention**:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- **Feedforward Network (MLP)**: Applies nonlinear transformations with GELU and dropout.
- **Residual Connections & Layer Norm**: Improve training stability.
- **Classification Head**: Uses a [CLS] token to summarize sequence-level information.

Figure 2.7 illustrates a deep Transformer architecture for ECG classification [70]. Notable variations include Liu et al.'s convolutional ViT for heart failure detection [43] and Zou et al.'s DWT-CNNTRN combining wavelet transforms with Transformer layers [86].
This work draws upon such models to incorporate Transformer encoders into a hybrid architecture for multi-label ECG classification.

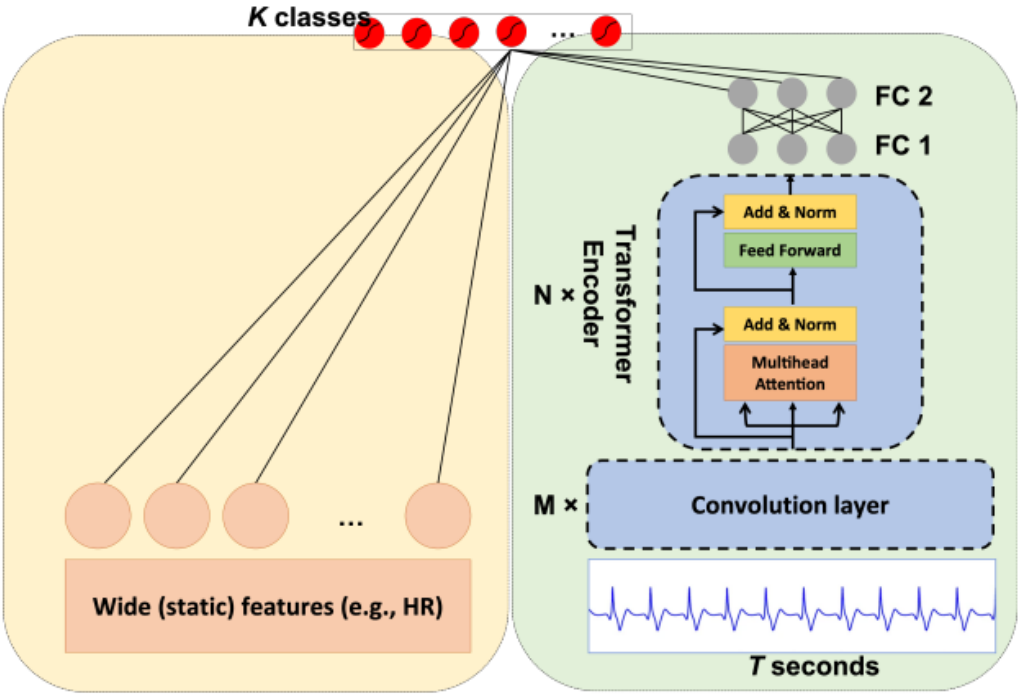

Figure 2.7: Architecture of the Wide and Deep Transformer Neural Network for 12-lead ECG classification, adapted from [70].

2.4.4 Attention Mechanism and Explainable AI (XAI)

Interpretability is essential in medical deep learning, particularly for tasks like ECG classification where incorrect decisions can have serious clinical implications. Beyond accuracy, models must offer insights into how predictions are made.
The attention mechanism, first introduced in neural machine translation [65], enhances both model performance and interpretability. It works by computing dynamic weights

that determine the importance of different input segments. The basic attention formula is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, and V are the query, key, and value matrices, and d_k is the dimension of the keys. This allows the model to focus on clinically significant features such as prolonged QRS complexes or abnormal ST-segment patterns.
There are several types of attention:

- **Self-Attention**: The input attends to itself, modeling intra-sequence dependencies. It is the core mechanism behind Transformer models.
- **Multi-Head Attention**: Extends self-attention by computing multiple attention heads in parallel, enabling the model to capture diverse patterns across different subspaces.
- **Cross-Attention**: Used to align information between two sequences, such as ECG leads and external signals.
- **Soft vs. Hard Attention**: Soft attention assigns continuous importance scores, while hard attention selects discrete input parts.

In addition to attention, explainable AI (XAI) techniques provide post-hoc interpretability by highlighting which input regions contributed most to a prediction. Notable methods include:

- **SHAP (SHapley Additive exPlanations)** [44]: Assigns importance scores to input features based on Shapley values from cooperative game theory.
- **LIME (Local Interpretable Model-Agnostic Explanations)**: Approximates the model locally with a simple surrogate model to explain individual predictions.
- **Grad-CAM (Gradient-weighted Class Activation Mapping)** [58]: Produces heatmaps by tracing gradients back to input regions most responsible for the output.

Figure 2.8 shows a Grad-CAM visualization applied to ECG signals, highlighting regions that heavily influenced the model's prediction [57]. These tools are essential for building clinical trust and verifying model reliability.
In this work, attention layers are embedded within the deep learning architecture to enhance the model's focus on diagnostically relevant waveform segments. Additionally, Grad-CAM is employed as an interpretability tool to visualize class-discriminative regions in ECG inputs.

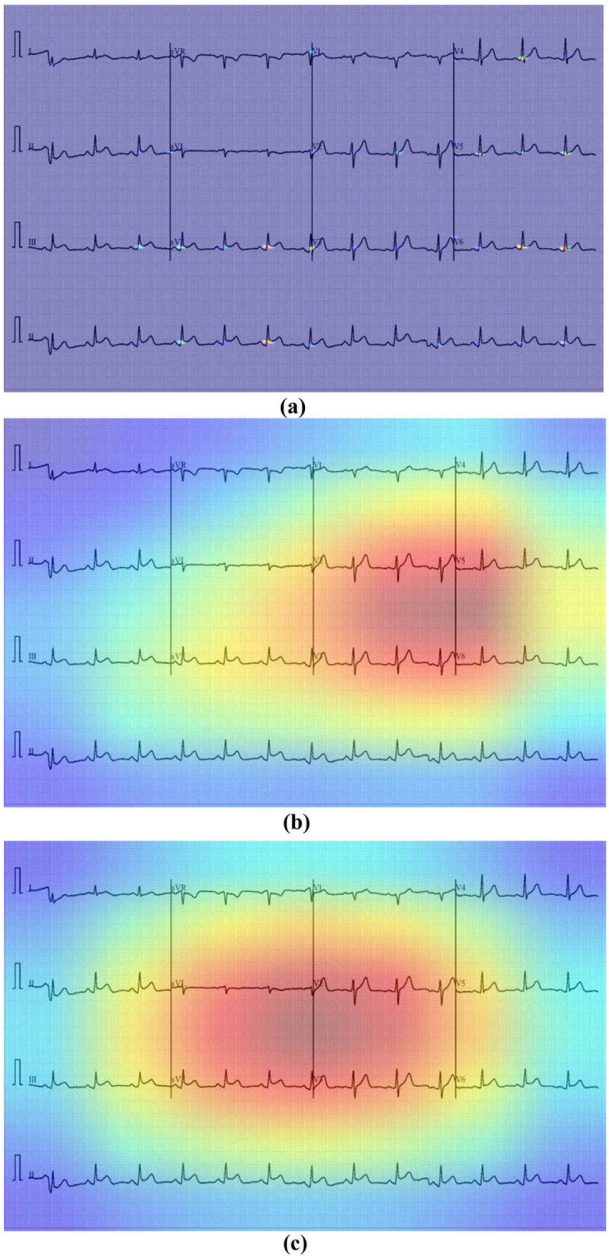

Figure 2.8: Grad-CAM visualization applied to ECG signals, highlighting regions that significantly influence the model's classification decisions [57].

2.5 Overview on Heart Diseases

This section presents an overview of the five cardiac conditions targeted in this study: Myocardial Infarction (MI), Normal ECG (NORM), ST/T Changes (STTC), Hypertrophy (HYP), and Conduction Disturbance (CD). These diagnostic categories, derived from the PTB-XL dataset [67], represent some of the most clinically significant and frequently encountered ECG labels. Understanding their characteristics is essential for the development of accurate and interpretable multi-label classification models [21].

2.5.1 Myocardial Infarction (MI)

Myocardial Infarction (MI) occurs when blood flow to a part of the heart is obstructed, leading to ischemic damage to the myocardium. It is typically classified into:

- **ST-Elevation Myocardial Infarction (STEMI)**: Complete coronary artery blockage, marked by ST-segment elevation on ECG.
- **Non-ST-Elevation Myocardial Infarction (NSTEMI)**: Partial blockage, typically shows ST depression or T-wave inversion [37].

ECG Characteristics [9]:

- STEMI: ST-segment elevation, pathological Q waves, T-wave inversion.
- NSTEMI: ST depression, T-wave inversion without ST elevation.

2.5.2 Normal ECG (NORM)

A normal ECG reflects healthy cardiac electrical activity and serves as the baseline for diagnosis.
ECG Characteristics [19, 21]:

- Sinus rhythm with heart rate between 60–100 bpm.
- P wave precedes each QRS complex.
- PR interval: 120–200 ms; QRS duration: <120 ms.
- ST segment is isoelectric; T waves upright in most leads.

2.5.3 ST/T Changes (STTC)

ST/T changes include a range of abnormalities in the ST segment and T wave, commonly indicating ischemia or repolarization defects [11].
ECG Characteristics:

- ST-segment depression or elevation.
- T-wave inversions or hyperacute T waves.

2.5.4 Hypertrophy (HYP)

Hypertrophy, particularly Left Ventricular Hypertrophy (LVH), results from chronic pressure overload such as hypertension, and leads to structural cardiac changes [39].
ECG Characteristics:

- Increased QRS voltage in precordial leads.
- ST-segment depression and T-wave inversion in lateral leads.
- Left axis deviation.

2.5.5 Conduction Disturbance (CD)

Conduction disturbances reflect delays or blocks in the heart's electrical conduction pathways, such as bundle branch blocks (BBB) [64].
ECG Characteristics:

- Prolonged QRS duration (>120 ms).
- RSR' pattern in V1 (Right BBB), or broad S waves in lateral leads (Left BBB).

Table 2.1: Summary of ECG features and clinical significance for each target class.

Class	**ECG Features**	**Clinical Significance**
MI	ST elevation, Q waves, T inversion	Cardiac ischemia and tissue death
NORM	Normal waveforms, sinus rhythm	Baseline healthy cardiac function
STTC	ST deviation, T-wave changes	Repolarization abnormalities, ischemia
HYP	High QRS voltage, LVH signs	Structural heart change due to overload
CD	Prolonged QRS, BBB patterns	Electrical conduction abnormalities

2.6 State of the Art on Multi-Label ECG Classification

Multi-label classification of electrocardiogram (ECG) signals has gained significant attention in recent years due to advances in deep learning (DL) techniques. Among available datasets, PTB-XL has become a benchmark resource for ECG research, offering high-resolution 12-lead recordings annotated with multiple co-occurring cardiac diagnoses. This section provides an overview of notable recent studies that applied various deep learning architectures for multi-label ECG classification using PTB-XL, with a focus on performance, explainability, and clinical applicability [6, 31, 52, 59, 62, 83].
Table 2.2 summarizes twelve representative studies that utilized PTB-XL (and comparable clinical datasets) for multi-label ECG classification.

Table 2.2: Summary of Prior Studies on Multi-Label ECG Classification Using PTB-XL dataset

Study	**Model**	**Dataset**	**Metrics**	**Advantages**	**Limitations**
Kang et al. (2025) [31]	xLSTM + STFT	PTB-XL	F1 = 72.3%, Accuracy = 89.4%	Captures time-frequency patterns	Lacks interpretability tools
Sethi et al. (2025) [59]	ProtoECGNet	PTB-XL	AUC = 91.2%	Prototype-based case interpretability	Sensitive to prototype selection
Prabhakararao et al. (2023) [52]	Temporal CNN with Attention	PTB-XL	F1-score = 76.51%	Integrates spatial-temporal attention	Moderate performance
Rawi et al. (2023) [54]	Inception, MobileNet + TPE	PTB-XL, Ningbo, Georgia	Accuracy = 97.89%, Precision = 90.83%	High performance across datasets	Performance varies across datasets
Strodthoff et al. (2020) [62]	ResNet / Inception	PTB-XL	AUC = 92.5%	Well-benchmarked on large dataset	No interpretability tools used
Huang et al. (2024) [27]	MRM-Net (Multi-Resolution)	PTB-XL, CPSC2018	F1 = 73.7%	Captures hierarchical features via multi-scale design	Slightly lower F1 than newer models
Kim et al. (2021) [33]	SE-ResNet (k-Labelsets)	PTB-XL	Accuracy = 93.2%	Models label dependencies explicitly	Risk of overfitting on small labels
Chen et al. (2023) [13]	Conditional Bayesian Net	PTB-XL	Label dependency modeling	Supports label hierarchy reasoning	High computational complexity
Zhang et al. (2023) [82]	CoT Attention + Multi-task Learning	PTB-XL	F1 = 83.3%, AUC = 91.3%	Multi-task support with interpretability	Complex design and training
Ibrahim et al. (2024) [29]	CNN + Transformer	PTB-XL	F1 = 81.2%, AUC = 91.8%	Combines spatial and global context features	High FLOPs, slow training
Yousef et al. (2024) [79]	CNN + Demographic Stratification	PTB-XL	Accuracy = 91–95%	Studies fairness across population groups	Generalizability limitations remain
Huang et al. (2023) [25]	CICST-DNN (Cost-sensitive)	PTB-XL	Weighted F1 = 85.1%	Handles imbalance with adaptive thresholds	Requires precise tuning

2.6.1 Critical Analysis

The reviewed studies present a diverse set of architectures and strategies for multi-label ECG classification using PTB-XL. While models like xLSTM-ECG [31] and ProtoECGNet [59] achieved reasonable performance, they either lacked interpretability mechanisms or relied on prototype selection strategies that are computationally expensive and sensitive to training noise.

Attention-based models such as ATCNN [52], CoT-Attention [82], and CNN+Transformer [29] demonstrated improvements in focusing on pathological waveform segments, but these approaches significantly increased model complexity (e.g., exceeding 700M FLOPs in some cases), limiting their applicability in real-time or resource-constrained clinical environments.

Some studies achieved high accuracy using optimization techniques (e.g., TPE in [54]), but their generalizability was often limited to the specific datasets used. Similarly, although ResNet/Inception [62] and MRM-Net [27] provided strong baselines, they lacked interpretability tools critical for clinical adoption.

Efforts to handle label imbalance and correlation—such as CICST-DNN [25] and SE-ResNet [33]—showed promising F1-scores, yet still struggled with performance degradation on underrepresented classes. Only a few studies explored hierarchical label dependencies explicitly, such as the work of Chen et al. [13], which used conditional Bayesian networks, though it incurred high computational cost.

Moreover, demographic fairness was rarely addressed; Yousef et al. [79] analyzed bias across population subgroups but did not report significant performance gains.

In summary, current state-of-the-art models often emphasize either performance or interpretability, but rarely both. Additionally, few models simultaneously address the key challenges of class imbalance, computational cost, interpretability, and generalizability. These gaps highlight the need for models that balance performance, interpretability, and efficiency. The hybrid architecture proposed in this study—combining CNNs, Transformers, and self-attention with integrated XAI tools—aims to address these challenges using only the PTB-XL dataset.

Conclusion

The review of recent deep learning approaches for multi-label ECG classification using the PTB-XL dataset reveals notable progress, yet several key challenges persist. Many studies achieved high classification performance, but often at the expense of interpretability, computational efficiency, or generalizability across diverse patient populations. Models such as ProtoECGNet and CICST-DNN have attempted to address interpretability and class imbalance, respectively, but few works succeeded in balancing all critical aspects simultaneously.

Moreover, inconsistencies in evaluation metrics—such as the selective reporting of macro vs. micro F1-scores, or omission of class-wise recall—hinder objective model comparison. Computationally intensive models, including Transformer-based and attention-heavy architectures, further raise concerns regarding real-time deployment in clinical environments.

In response to these limitations, this project explores a hybrid deep learning model that combines CNNs, LSTMs, and Transformers, along with attention mechanisms and Grad-CAM, aiming to improve multi-label ECG classification on the PTB-XL dataset while maintaining a balance between performance and interpretability.

Chapter 3

Proposed Method

3.1 Introduction

While ECG signal classification plays a vital role in the early detection of cardiac conditions, it presents several challenges due to its multi-label nature, inter-patient variability, and overlapping waveform morphologies. To address these complexities, this chapter outlines the model architectures and training strategies implemented for automated ECG classification.
The first section focuses on deep learning architectures—including CNN, LSTM, and Transformer models—along with hybrid configurations that combine their complementary strengths. The second section introduces traditional machine learning classifiers such as Random Forest, XGBoost, and SVM.
For each model, we describe its architecture, training setup, and the rationale behind the design choices. Evaluation results and comparative metrics are presented in the next chapter.

3.2 Contribution 1 :Comparative Study of Deep and Traditional Approaches for ECG Classification

3.2.1 Data Preprocessing and Input Format for All Models

All models—deep learning and conventional machine learning—were trained and evaluated on the same subset of 10,000 ECG recordings from the PTB-XL dataset. Each sample corresponds to a 12-lead electrocardiogram signal recorded over 10 seconds with a sampling rate of 100 Hz, resulting in an input matrix of shape 1000×12. Here, the 1000 time steps represent voltage measurements taken at 100 Hz, and the 12 columns correspond to the standard clinical ECG leads (e.g., I, II, III, aVR, aVL, aVF, V1–V6).
Raw waveforms were loaded directly from WFDB files and transformed into NumPy arrays to facilitate model input. Z-score normalization was applied independently to each channel to standardize the signal distribution. Diagnostic labels were extracted from SCP-coded statements and encoded into binary multi-label vectors using the `MultiLabelBinarizer` from `scikit-learn`.
The dataset was split using stratified sampling to preserve label distribution, with 80% of the samples used for training and 20% for testing. A fixed random seed was used to ensure reproducibility across experiments. All models, unless otherwise specified, used

the same preprocessed signals as input, without additional filtering, transformation, or handcrafted feature extraction.
This unified preprocessing pipeline ensures consistency across models and enables a fair comparison of their performance under identical experimental conditions.

Conventional Machine Learning Models

3.2.2 Multi-Layer Perceptron (MLP)

In our ECG classification task, an MLP was employed to process preprocessed, flattened ECG signals and predict diagnostic categories.
We applied a 3-layer MLP to the dataset as follows:

- **Input layer:** size equals the number of features in the flattened input data (d = input_size).
- **Hidden layers:**
  - First hidden layer with 128 neurons, using ReLU activation.
  - Second hidden layer with 64 neurons, also using ReLU.
- **Output layer:** Number of neurons equals the number of classes (num_classes), followed by a softmax to output class probabilities.

Forward Pass Computation

For a single input sample $\mathbf{x} \in \mathbb{R}^d$:

$$\begin{aligned} \mathbf{z}_1 &= \mathbf{W}_1\mathbf{x} + \mathbf{b}_1, \quad \mathbf{h}_1 = \text{ReLU}(\mathbf{z}_1) \\ \mathbf{z}_2 &= \mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2, \quad \mathbf{h}_2 = \text{ReLU}(\mathbf{z}_2) \\ \mathbf{z}_3 &= \mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3, \quad \hat{\mathbf{y}} = \text{softmax}(\mathbf{z}_3) \end{aligned}$$

Activation Functions

- **ReLU (Rectified Linear Unit):**

$$\text{ReLU}(x) = \max(0, x)$$

  This introduces non-linearity by zeroing out negative values.

- **Softmax:** Converts logits into probabilities over C classes:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}$$

  Ensures the output probabilities sum to 1.

Training Details

- **Loss function:** Cross-entropy between predicted $\hat{\mathbf{y}}$ and true class labels $\mathbf{y}$:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

  where C is the number of classes.

- **Optimizer:** Adam with learning rate 0.001.

Data Preparation

- Inputs $\mathbf{X}_{\text{train}}$ and $\mathbf{X}_{\text{test}}$ are converted to tensors .
- Target labels $\mathbf{y}_{\text{train}}$ and $\mathbf{y}_{\text{test}}$ are converted to class indices for `CrossEntropyLoss`.

Hyperparameters and Model Settings Table

Hyperparameter	Value
Hidden Layer 1 size	128
Hidden Layer 2 size	64
Activation function	ReLU
Output activation	Softmax
Loss function	CrossEntropyLoss
Optimizer	Adam
Learning rate	0.001

Table 3.1: MLP Hyperparameters and Settings

3.2.3 Random Forest

How It Works (Specific to Our ECG Classification Task)

In our project, we applied a **Random Forest classifier** to predict multiple diagnostic categories related to cardiac conditions from ECG signal data. The model operates by constructing a large number of decision trees from feature representations of raw ECG signals and combining their outputs for robust classification.

Each ECG record in the PTB-XL dataset is represented as a multi-lead time series (e.g., 12-lead signals). These signals were first preprocessed and **flattened** into a single feature vector per sample. Thus, each instance $x_i \in \mathbb{R}^d$ corresponds to the raw voltage values concatenated across all leads and time steps, forming a high-dimensional vector. This flattened vector serves as input to the Random Forest model, where each element becomes a candidate feature for node splits in the decision trees.

During training, `RandomForestClassifier` employs **bootstrap sampling**. For each of the $T = 100$ trees, a random subset of the training data is sampled **with replacement**. This means some samples may be duplicated, while others may be excluded from a tree's training set. Although we did not use it, these excluded samples can serve for out-of-bag (OOB) error estimation. This sampling strategy introduces variation across trees and enhances generalization.

At every internal node of each tree, a random subset of features is selected—specifically $\sqrt{d}$ features, as determined by the setting `max_features='sqrt'`. From this subset, the algorithm identifies the best feature and threshold that minimize impurity. The impurity is measured using the Gini impurity, defined as:

$$\text{Gini}(t) = 1 - \sum_{k=1}^{K} p_k^2$$

where K is the number of target classes and p_k is the proportion of samples of class k at node t. The chosen split aims to increase class purity in the resulting child nodes.

Each decision tree is grown recursively until one of several stopping conditions is met: the tree reaches a maximum depth of 10, the number of samples at a node is less than `min_samples_split = 2`, or a child node contains fewer than `min_samples_leaf = 4` samples. These constraints are applied to control model complexity and prevent overfitting.

At inference time, each trained tree $h_i(x)$ independently predicts a class label for a new ECG sample x. The final model output is determined through **majority voting**, defined as:

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \ldots, h_T(x)\}$$

This ensemble approach increases prediction stability and reduces sensitivity to noise or outliers in the signal data.

To handle class imbalance in the dataset—since some disease categories are significantly underrepresented—we used `class_weight='balanced'`. This option adjusts the weight of each class in the impurity calculation so that rare classes are given greater importance during training. As a result, the model is less likely to ignore minority classes and achieves better balance in multi-class classification performance.

The parameters used are summarized in Table 3.2.3.

Parameters Used.

Parameter	**Value**
Number of Trees (`n_estimators`)	100
Max Tree Depth (`max_depth`)	10
Min Samples Split (`min_samples_split`)	2
Min Samples Leaf (`min_samples_leaf`)	4
Max Features per Split (`max_features`)	`sqrt`
Bootstrap Sampling (`bootstrap`)	True
Class Weighting (`class_weight`)	`balanced`
Splitting Criterion (`criterion`)	`gini (default)`
Number of Features	∼12,000

Table 3.2: Parameters used in Random Forest model

3.2.4 XGBoost-Based Model

In this study, XGBoost was employed as a powerful gradient boosting framework to classify ECG signals based on disease type, Figure 3.1 provides a visual representation of the XGBoost architecture. The dataset used consists of 12-lead ECG recordings from the PTB-XL dataset, with each signal flattened into a fixed-length one-dimensional vector to serve as the input feature matrix. Formally, the input data $X_{\text{train\_flat}}$ and $X_{\text{test\_flat}}$ belong to $\mathbb{R}^{n \times d}$, where n is the number of samples and d is the number of features per sample, derived from the raw signal. The corresponding target labels are represented as one-hot encoded vectors, indicating the presence or absence of each disease class in a multi-class classification setting.

The model architecture follows the classic gradient boosting framework, where an ensemble of $M = 200$ decision trees (`n_estimators`) is constructed sequentially, illustrated in table 3.3 . Each tree in the sequence is trained to correct the classification errors made by the previous ensemble using the gradient and Hessian of the multi-class log-loss. The model is updated iteratively as:

$$F_m(x_i) = F_{m-1}(x_i) + \eta f_m(x_i)$$

where $\eta = 0.06$ (`learning_rate`) controls the contribution of each tree to the final prediction, effectively shrinking step size to improve generalization and reduce overfitting.

The trees are restricted to a maximum depth of 6 (`max_depth`) to avoid overfitting, and randomness is introduced both at the level of samples (`subsample` $= 0.8$) and features (`colsample_bytree` $= 0.8$) to promote model diversity. Additionally, a regularization term $\gamma = 1$ is applied to limit unnecessary splits in trees, where a split is only performed if it reduces the loss by at least this threshold.

To further mitigate overfitting in this noisy biomedical signal classification task, L1 (`reg_alpha` $= 0.1$) and L2 (`reg_lambda` $= 1$) regularization penalties were applied. These terms respectively encourage sparsity in feature selection and penalize overly complex models. Figure 3.1 provides a visual representation of the XGBoost architecture.

In the context of ECG data, this approach is particularly effective because it can handle high-dimensional inputs and learn complex decision boundaries while incorporating regularization techniques to control overfitting—a critical requirement given the noise and variability often present in biomedical signals. XGBoost's iterative nature enables it to focus on correcting misclassified ECG samples, progressively improving classification performance across all disease categories.

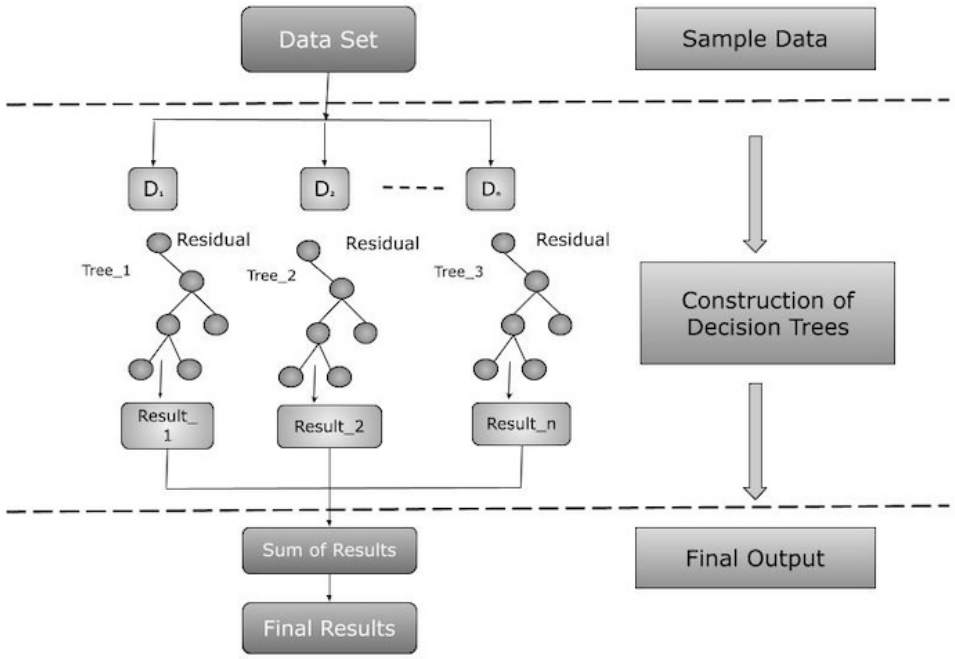

Figure 3.1: XGBoost architecture [1]

3.2.5 Core Hyperparameters Used in XGBoost

Parameter	Value	Description
`n_estimators`	200	Number of boosting rounds (trees in the ensemble).
`learning_rate`	0.06	Step size shrinkage used to prevent overfitting.
`max_depth`	6	Maximum depth of each decision tree. Controls model complexity.
`subsample`	0.8	Fraction of training samples used for growing each tree.
`colsample_bytree`	0.8	Fraction of features randomly sampled for each tree.
`gamma`	1	Minimum loss reduction required to make a further partition on a leaf node.
`reg_alpha`	0.1	L1 regularization term on weights. Encourages sparsity.
`reg_lambda`	1	L2 regularization term on weights. Controls overfitting.
`scale_pos_weight`	1	Balances the positive and negative weights in unbalanced datasets.

Table 3.3: Core hyperparameters of the XGBoost model and their values used during training

3.2.6 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) were utilized to classify multi-class ECG signals from the PTB-XL dataset. Given the high dimensionality of the ECG recordings—where each 12-lead signal was flattened into a one-dimensional vector—Principal Component Analysis (PCA) was first applied to reduce dimensionality while preserving 95% of the dataset's variance. This corresponded to retaining only 4 principal components, reducing the input matrix $X \in \mathbb{R}^{n \times d}$ to $X_{\text{PCA}} \in \mathbb{R}^{n \times 4}$.
After dimensionality reduction, a linear SVM model was trained using `LinearSVC` with a maximum of 5,000 iterations and `class_weight='balanced'` to mitigate the effects of class imbalance. Since `LinearSVC` does not inherently produce probabilistic outputs, Platt scaling was applied using sigmoid calibration:

$$P(y = k \mid x) = \frac{1}{1 + \exp(Af_k(x) + B)}$$

This calibration was implemented using `CalibratedClassifierCV` with 3-fold cross-validation to produce reliable posterior probability estimates.
To handle the multi-class classification task, the SVM model was embedded within a One-vs-Rest (OvR) strategy, implemented via `OneVsRestClassifier`. This approach involved training K binary classifiers—each focused on distinguishing one class from the rest.
The model training and evaluation flow was as follows:

- `fit(X_train_pca, y_train)` — trained the classifier using the reduced 4-dimensional feature set.
- `predict(X_test_pca)` — generated predicted class labels.
- `predict_proba(X_test_pca)` — produced class probability distributions.

The classifier's performance was evaluated using appropriate metrics for multi-class imbalanced datasets. Additionally, scatter plots were used to qualitatively assess prediction quality by comparing predicted and true labels across sample indices.
From a computational perspective, each prediction required approximately 13,610 floating point operations (FLOPs) as shown in table 3.4 , and the model had an estimated 6,805 parameters. The total number of support vectors across all OvR classifiers was approximately 1,701, reflecting the complexity of the learned decision boundaries in the PCA-reduced feature space.
The complete SVM-based ECG classification pipeline is summarized below:

1. **Raw ECG signal → PCA**: Dimensionality reduction (4 components retained).
2. **PCA → Linear SVM (OvR)**: Classifier training with balanced class weighting and max 5,000 iterations.
3. **Decision scores → Probabilities**: Sigmoid-based Platt scaling with 3-fold calibration.
4. **Probabilities → Predictions**: Final prediction via OvR strategy.

Parameter	Value
PCA `n_components`	0.95 (4 components)
SVM kernel	Linear (`LinearSVC`)
Class weight	Balanced
Max iterations	5000
Calibration method	Sigmoid (Platt scaling)
Cross-validation folds	3
Multiclass strategy	One-vs-Rest
Number of support vectors	∼1,701
Number of features	4
Estimated parameters	6,805
FLOPs per prediction	13,610

Table 3.4: Summary of Parameters Used in the One-vs-Rest Calibrated Linear SVM

3.2.7 Ensemble Learning Model

We used a `VotingClassifier` ensemble that combines Random Forest and XGBoost models with soft voting. The ensemble averages the predicted probabilities of each class from both models and selects the class with the highest average probability.
The ECG data, previously flattened into a two-dimensional format, contains many features. To reduce dimensionality and improve performance, we applied feature selection using `SelectKBest` with the ANOVA F-test (`f_classif`), selecting the top 100 most informative features. This step helps to speed up training and focus the models on the most relevant data aspects.
For Random Forest, we used 100 trees with a maximum depth of 5 to prevent overfitting, and fixed the random state to 42 for reproducibility. The XGBoost model was configured with 100 boosting rounds, a maximum tree depth of 5, and a learning rate of 0.1. It outputs class probabilities using the `multi:softprob` objective and optimizes multiclass log-loss. Label encoding warnings were disabled, and the random state was set to 42.
The `VotingClassifier` combines these two models using soft voting, allowing the ensemble to leverage the robustness of Random Forest and the error-correcting strength of XGBoost. After training, the ensemble generates predicted labels and class probability distributions for the test data, which can be used for evaluation and further analysis.
A summary of the key parameters is presented in Table 3.5.
Parameters Summary

Component	Parameter	Value
Random Forest	n_estimators	100
	max_depth	5
	random_state	42
XGBoost	n_estimators	100
	max_depth	5
	learning_rate	0.1
	objective	multi:softprob
	eval_metric	mlogloss
	use_label_encoder	False
	random_state	42
VotingClassifier	voting	soft
Feature Selection	k (top features)	100

Table 3.5: Summary of Parameters Used in the Ensemble Model

3.3 Deep Learning Models

3.3.1 CNN-Based Model

Convolutional Neural Networks (CNNs) have shown strong performance in biomedical time-series analysis due to their ability to capture spatially localized patterns [**?**]. In this work, a 1D CNN model was designed to classify 12-lead ECG signals of fixed length (1000 time steps) extracted from the PTB-XL dataset.
The architecture consists of three sequential convolutional blocks:
The first block uses a Conv1D layer with 64 filters and a kernel size of 5. This relatively small number of filters reduces computational cost while still allowing the network to extract basic morphological features such as P-waves or initial QRS shapes. The kernel size of 5 was selected to provide a balance between local detail and contextual information over small temporal windows.
The second block increases the capacity with 128 filters and a similar kernel size (5), enabling the model to extract higher-level abstractions from the previously learned representations.

The third block retains 128 filters but reduces the kernel size to 3, allowing the model to focus on finer-grained details and improve resolution near sharp transitions (e.g., QRS peaks). Each block includes BatchNormalization to stabilize learning, followed by MaxPooling1D to reduce dimensionality and control overfitting. Dropout layers (with rates of 0.3 and 0.4) are employed to enhance generalization.

Following the convolutional blocks, a GlobalAveragePooling1D layer is used to aggregate temporal features across the entire sequence, reducing the model size compared to Flatten while preserving important global characteristics. This is followed by a dense layer of 256 ReLU-activated units, which introduces non-linearity and enables learning complex combinations of extracted features.

The final output layer consists of 5 sigmoid-activated units, corresponding to the five diagnostic classes in a multi-label classification setup.

Training Configuration The model was trained using the Adam optimizer with an initial learning rate of 0.0005, chosen for its adaptive learning capabilities and robustness in noisy gradients. The loss function used is Focal Loss [42], with $\gamma = 2.0$ and $\alpha = 0.25$, which helps to address class imbalance by down-weighting easy samples and focusing the model's learning on hard-to-classify examples — a common challenge in ECG datasets.

To prevent overfitting, EarlyStopping was applied based on the validation AUC, stopping training after 10 epochs without improvement. Additionally, ModelCheckpoint was used to save the best model weights during training.

A batch size of 32 was selected to strike a balance between GPU memory constraints and convergence stability. Training was performed for a maximum of 100 epochs, with actual duration controlled dynamically via early stopping.

Justification of Design Choices The use of three convolutional layers allows the model to progressively extract low, mid, and high-level features, striking a balance between expressive power and risk of overfitting. Deeper CNNs tend to capture more complex hierarchies, but beyond three layers, gains were marginal.

The progressive increase in filter numbers from 64 to 128 enables the network to learn increasingly abstract representations as the depth increases, without introducing a sudden spike in computational complexity.

The choice of kernel sizes—5 in the initial layers and 3 in the final one—enables the model to detect both broader waveform structures such as T-waves and finer temporal transitions such as QRS onsets.

GlobalAveragePooling1D was selected over Flatten to reduce the number of trainable parameters and prevent overfitting, while still capturing the essential temporal features over the full sequence.

ReLU was used in the hidden layers due to its efficiency in training deep networks and resistance to vanishing gradients. The sigmoid activation in the output layer is suited for multi-label classification, as it produces independent probability estimates for each class. Figure 3.2 illustrates the convolutional layers and pooling operations used to extract spatial features from the ECG signals.

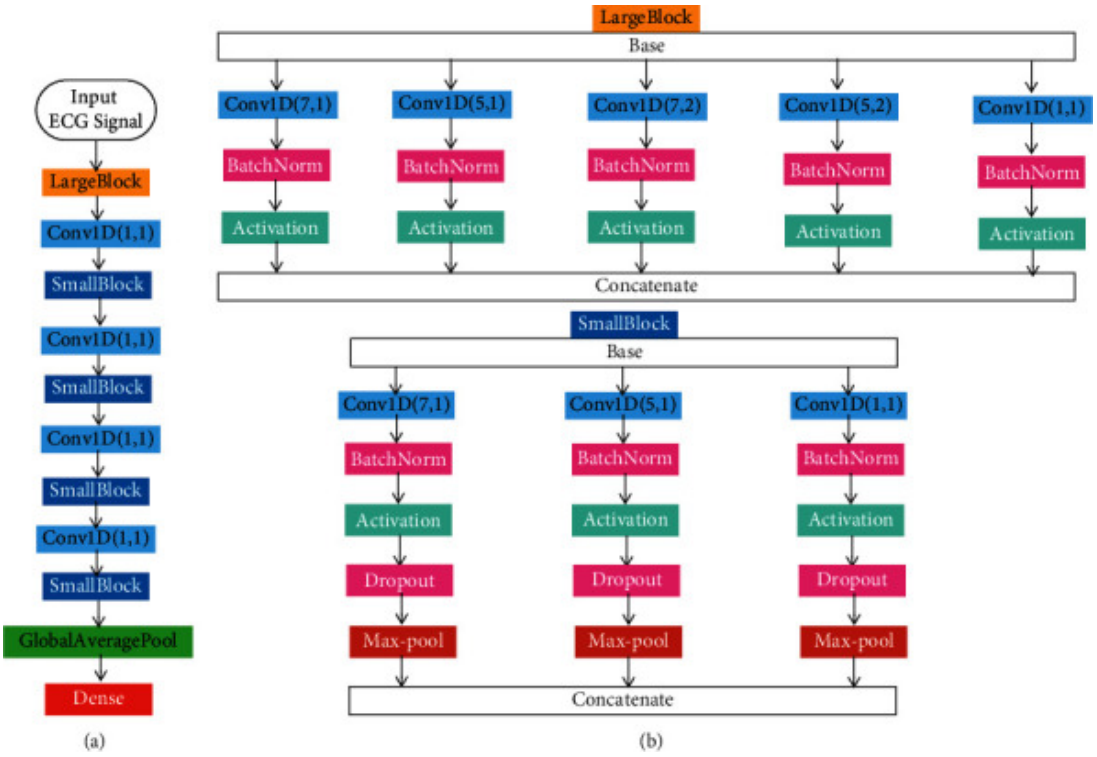

Figure 3.2: 1D CNN architecture used for multi-label classification of 12-lead ECG signals, adapted from [71].

3.3.2 LSTM-Based Model

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) specifically designed to handle long-range temporal dependencies by mitigating the vanishing gradient problem [24]. Given the temporal nature of ECG signals, LSTM architectures are particularly suitable for modeling sequential cardiac patterns that span across multiple heartbeats.

In this work, a two-layer LSTM model was developed to classify multi-label ECG recordings with input dimensions of 1000×12, representing 12-lead signals over 1000 time steps. The architecture starts with an LSTM layer consisting of 128 units and configured with `return_sequences=True`, which preserves the temporal structure across the full sequence to be processed by the subsequent layer. This setting is crucial for maintaining rich temporal dynamics between early and later timesteps.

The second LSTM layer has 64 units and outputs a fixed-size latent vector summarizing the full sequence. Both layers are followed by `BatchNormalization` to stabilize training and improve convergence. Dropout regularization is applied after each layer (with rates of 0.3 and 0.4, respectively) to prevent overfitting.

The encoded temporal features are passed through a fully connected `Dense` layer with 256 ReLU-activated units, enabling the model to learn complex high-level representations. The final output layer consists of 5 neurons with `sigmoid` activation to produce independent probability scores for each diagnostic class in the multi-label classification setup.

Training Configuration The model was compiled using the Adam optimizer with an initial learning rate of 0.0005, known for its adaptive learning rate and efficiency in training deep models. To address class imbalance and guide learning toward difficult examples, the Focal Loss [42] was employed with parameters $\gamma = 2.0$ and $\alpha = 0.25$.

Training was monitored using the validation AUC metric. `EarlyStopping` was triggered if no improvement was observed over 10 consecutive epochs, and `ModelCheckpoint` was used to store the weights of the best-performing model during training.

A batch size of 32 was chosen as a compromise between gradient estimation quality and memory efficiency. The maximum number of training epochs was set to 100, but training duration was dynamically adjusted based on early stopping.

Justification of Design Choices Using two LSTM layers enables hierarchical modeling of temporal dependencies: the first captures local dynamics, while the second summarizes the entire sequence into a compact feature vector.

The hidden sizes of 128 and 64 were selected based on empirical experiments; larger units increased computational complexity with no significant gain, while fewer units degraded performance.

BatchNormalization was introduced after each LSTM layer to reduce internal covariate shift and improve training stability. Dropout with gradually increasing rates was used to mitigate overfitting at deeper layers.

The fully connected layer with 256 ReLU units provides a non-linear transformation of the learned features before classification. ReLU is effective in accelerating convergence and reducing vanishing gradients, while sigmoid activation in the output layer ensures that each class is treated independently—suitable for multi-label scenarios.

The internal structure of the LSTM cell used for modeling temporal dependencies is depicted in Figure 3.3.

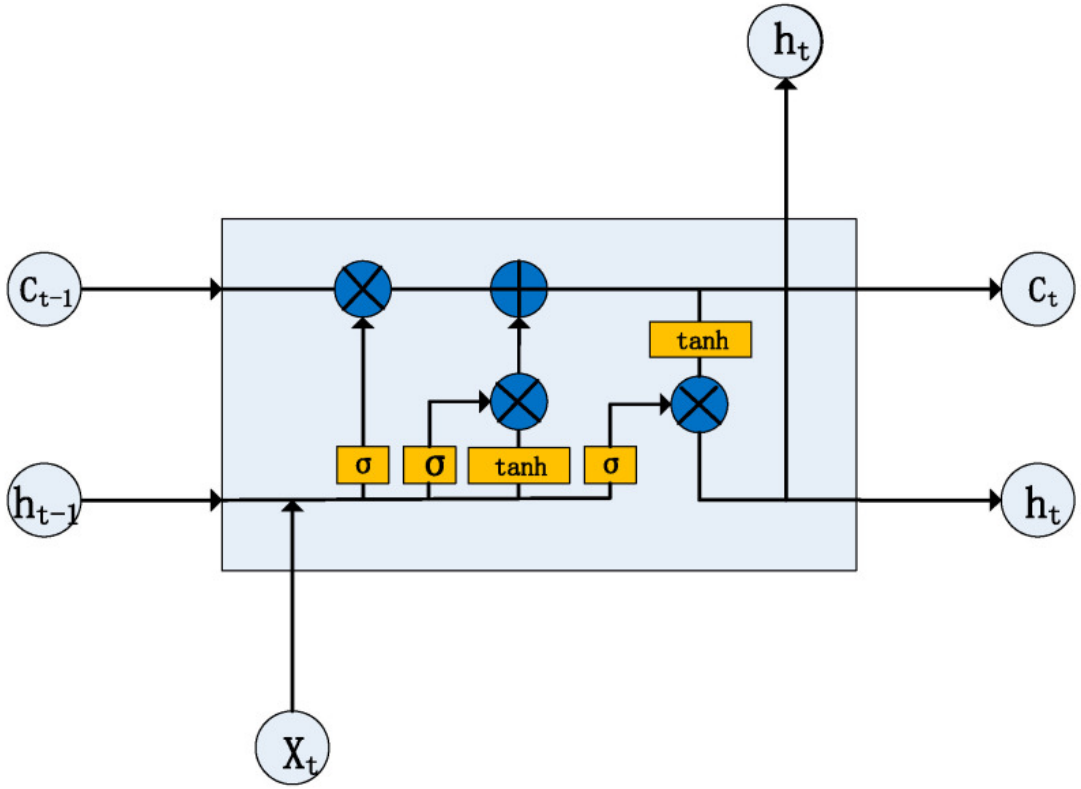

Figure 3.3: Structure of the LSTM cell used for ECG signal classification, adapted from [5].

3.3.3 Transformer-Based Model

Transformer architectures, initially developed for natural language processing, have recently gained attention in biomedical time-series modeling due to their ability to capture long-range temporal dependencies using self-attention mechanisms [65]. In this study, a pure Transformer-based architecture was developed for multi-label classification of 12-lead ECG signals, each represented as a sequence of 1000 time steps.

The architecture consists of two stacked Transformer encoder blocks, each designed to process the full temporal context of the ECG sequence:

Each encoder block begins with a `MultiHeadAttention` layer configured with 4 heads and a key dimension of 64. This allows the model to learn relationships between different time points in the signal simultaneously across multiple subspaces.

A `Dropout` layer (rate = 0.2) follows to regularize the attention weights and prevent overfitting. A residual connection is added between the input and attention output, followed by `LayerNormalization` to stabilize and accelerate training.

Then, a feed-forward sublayer is applied: a `Dense` layer with 128 ReLU-activated units, followed by another dense layer that projects back to the original input dimension. Another residual connection and `LayerNormalization` layer follow, completing the encoder block.

After the two encoder layers, a `GlobalAveragePooling1D` layer aggregates the sequence-level features into a fixed-length vector. This is followed by a fully connected layer with

256 units and ReLU activation, and a final `Dropout` layer (rate = 0.5) to mitigate overfitting. The output layer consists of 5 `sigmoid`-activated neurons, corresponding to the five diagnostic classes in a multi-label setting.

The model was trained using the Adam optimizer with an initial learning rate of 0.0005. The binary cross-entropy loss function was employed, which is well suited for multi-label classification tasks where each class prediction is independent.

To enhance training efficiency and prevent overfitting, two callbacks were used:
`EarlyStopping` monitored validation AUC and halted training if no improvement was observed for 10 consecutive epochs.
`ModelCheckpoint` saved the weights of the model that achieved the highest validation AUC.
A batch size of 32 was used during training, and the maximum number of epochs was set to 100, with training typically stopping early based on validation performance.

The use of two encoder blocks provides a sufficient balance between model capacity and training stability. Deeper configurations were avoided to prevent overfitting, especially given the relatively small dataset size compared to language applications.

Multi-head attention with 4 heads and key dimension 64 enables the model to learn diverse temporal relationships and focus on different waveform structures (e.g., ST segment, T-wave).

The feed-forward layers with 128 units were empirically found to be sufficient for abstracting representations while keeping the parameter count manageable.

GlobalAveragePooling1D was preferred over flattening to reduce the number of parameters and retain global temporal patterns without spatial explosion.

ReLU activation in hidden layers accelerates convergence, and sigmoid activation in the output layer ensures proper multi-label probability estimation.
Figure 3.4 presents the Transformer encoder blocks employed to capture global temporal relationships in the ECG sequences.

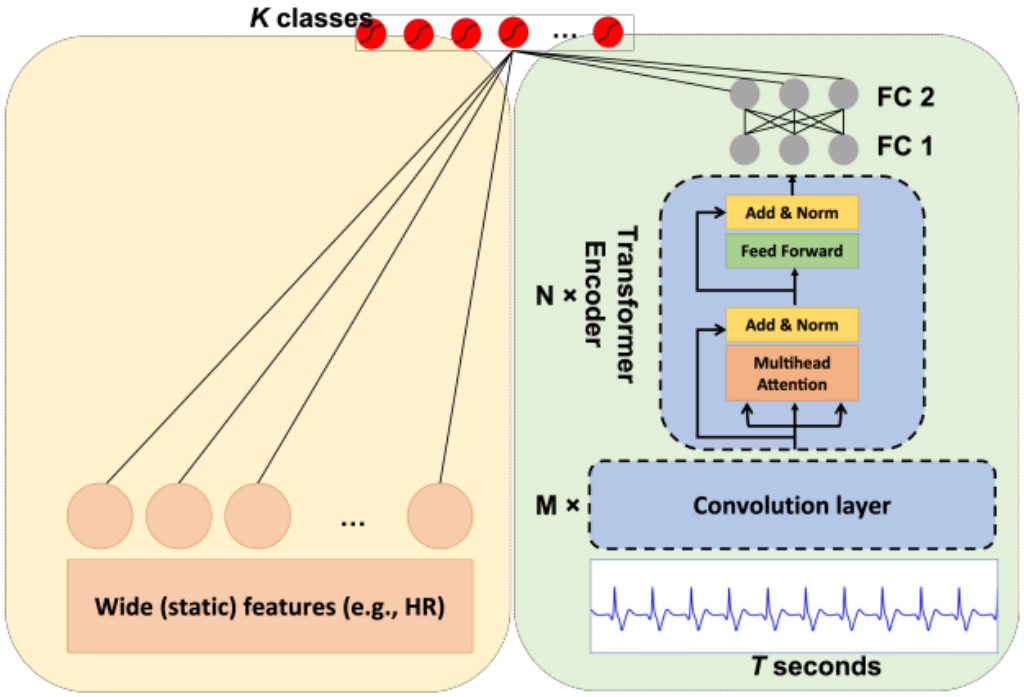

Figure 3.4: Transformer-based architecture for multi-label classification of 12-lead ECG signals, adapted from [41].

3.3.4 CNN + LSTM Hybrid Model

To combine the spatial learning strengths of convolutional layers with the temporal modeling capabilities of recurrent networks, a hybrid CNN + LSTM architecture was designed for multi-label ECG classification using 12-lead signals of length 1000 samples from the PTB-XL dataset. This integration was motivated by the complementary nature of both components: CNNs are highly effective in extracting local morphological features from raw ECG signals (e.g., P-wave shapes, QRS complexes), while LSTMs excel in modeling temporal dependencies and sequential relationships across time. By combining both, the architecture is capable of capturing low-level spatial features and their higher-level temporal dynamics, which are critical for detecting overlapping or co-occurring cardiac conditions in real-world settings.

The architecture begins with a Conv1D layer using 64 filters and a kernel size of 5, aimed at capturing localized patterns such as P-waves and the onset of QRS complexes. A relatively small number of filters in this stage keeps the computational cost low while retaining essential features. The convolutional output is normalized via `BatchNormalization` and reduced in temporal resolution through `MaxPooling1D` (pool size = 2). A `Dropout` layer with a rate of 0.3 is applied to enhance generalization and reduce overfitting.

The extracted spatial features are then passed to two stacked LSTM layers:

The first LSTM layer has 64 units and uses `return_sequences=True`, which preserves the temporal dimension for the next LSTM layer. This layer models intermediate temporal dependencies across the ECG waveform.

The second LSTM layer has 32 units and compresses the sequence into a fixed-length vector that encapsulates the overall temporal dynamics. A second `Dropout` layer (rate = 0.3) follows to prevent co-adaptation of units.

The output is then passed through a dense layer of 256 units with ReLU activation, enabling non-linear transformation of the combined spatiotemporal representation. A final `Dropout` layer with a higher rate (0.5) is applied to mitigate overfitting, especially due to the high capacity of the dense layer.
The final output layer contains 5 units with sigmoid activation, suitable for multi-label classification, where each unit predicts the presence or absence of a specific diagnostic label independently.
Training Configuration. The model was compiled using the Adam optimizer with a learning rate of 0.0005. The Focal Loss [42] with parameters $\gamma = 2.0$ and $\alpha = 0.25$ was employed to address the issue of class imbalance, which is common in clinical ECG datasets. This loss emphasizes harder samples by down-weighting easy examples.
To prevent overfitting and retain optimal model weights, two callbacks were used:

- `EarlyStopping` was configured to monitor validation AUC and terminate training if no improvement was observed for 10 epochs.
- `ModelCheckpoint` saved the weights corresponding to the best validation AUC.

The model was trained for up to 100 epochs using a batch size of 32, which balances memory usage and convergence stability. Training was performed with validation monitoring to adaptively stop at the optimal epoch.
Justification of Design Choices. Combining CNN and LSTM enables the model to first capture spatially-localized morphological features (via CNN) and then learn their temporal progression (via LSTM), reflecting real physiological processes in ECG signals.

Using 64 filters in the CNN layer balances model complexity and capacity to detect early waveform features. Increasing this significantly showed overfitting without performance gain.

LSTM units set to 64 and 32 allow hierarchical temporal modeling. Deeper LSTM stacks were avoided due to convergence instability and increased training time.

ReLU activation is used in the dense layer for faster training and to avoid vanishing gradients. Sigmoid activation in the output layer is suitable for independent multi-label outputs.

Global dropout strategy (with rates from 0.3 to 0.5) was carefully applied after each core module to reduce overfitting without harming the learning capacity.

This hybrid model proved especially effective in learning both morphological and temporal aspects of ECG signals, making it suitable for capturing complex cardiac patterns in multi-label classification tasks.
The integration of convolutional feature extraction with sequential LSTM modeling is shown in Figure 3.5.

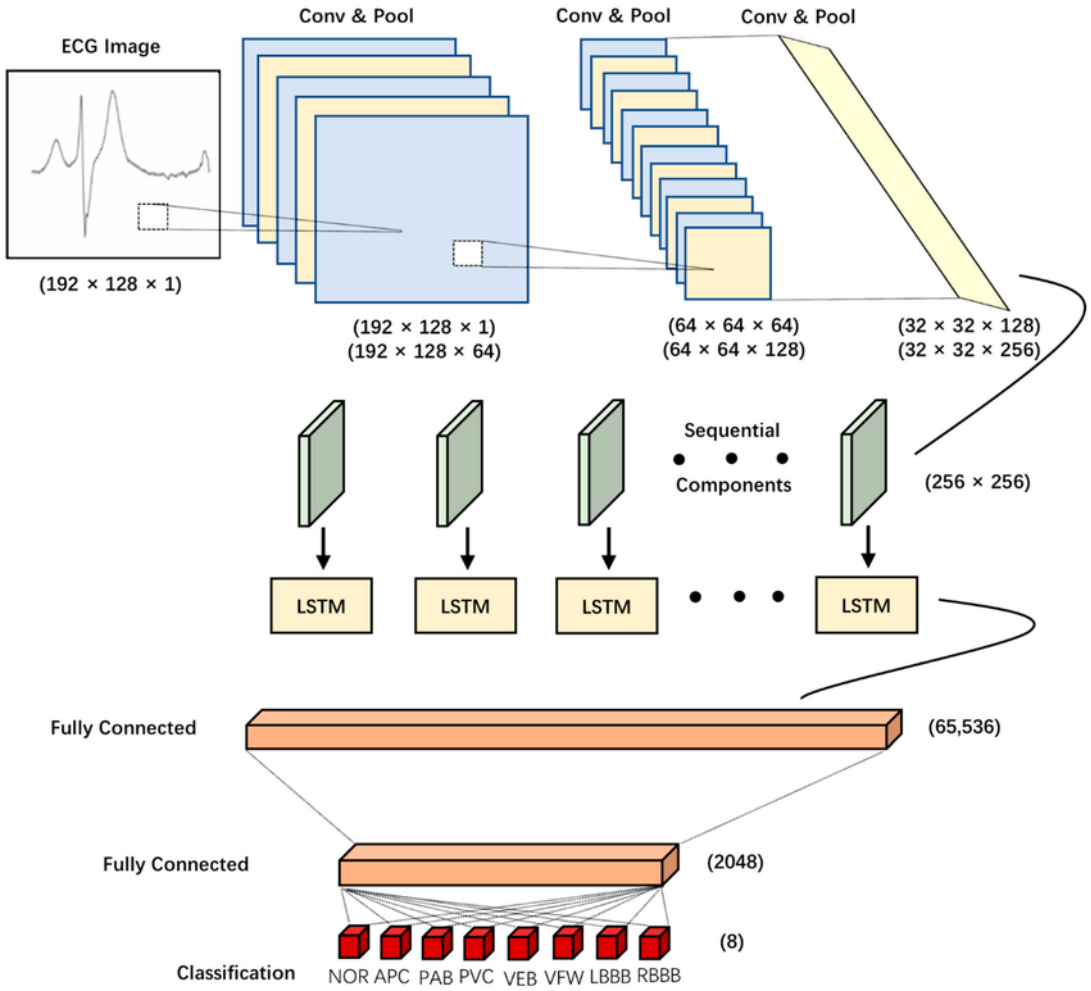

Figure 3.5: Proposed CNN-LSTM architecture for arrhythmia classification using 12-lead ECG signals, adapted from [84].

3.3.5 CNN + Transformer Hybrid Model

To combine the benefits of local and global feature extraction in ECG signal classification, a hybrid architecture integrating Convolutional Neural Networks (CNN) with Transformer encoders was designed. This model leverages CNNs for low-level morphological pattern detection and Transformers for capturing long-range temporal dependencies through self-attention mechanisms.
The input to the model consists of 12-lead ECG signals, each of shape (1000×12).

The network begins with a Conv1D layer comprising 64 filters and a kernel size of 3. This layer extracts localized features—such as the onset and peaks of cardiac waves—over short temporal windows. The kernel size was chosen as a compromise between local detail and computational efficiency. Following this, BatchNormalization is applied to stabilize learning and speed up convergence, and a Dropout layer (rate = 0.2) is used to improve generalization and reduce overfitting risk.

The resulting feature maps are passed into a Transformer encoder block, which consists of:

A MultiHeadAttention layer with 4 attention heads and key dimension 32. This compo-

nent enables the model to attend to multiple positions in the sequence simultaneously, capturing long-range dependencies that are critical in ECG signals (e.g., ST-T interactions, delayed repolarization).

Residual connections followed by LayerNormalization stabilize the gradient flow and ensure better training dynamics.

A feed-forward subnetwork (FFN) implemented as a Sequential model contains:
A dense layer with 128 ReLU-activated units.
A projection layer to match the input dimensionality.

This FFN allows for richer non-linear transformations over the attention output. A second residual connection and normalization complete the encoder block.

After feature encoding, the temporal features are aggregated via a GlobalAveragePooling1D layer, compressing the sequence into a fixed-size vector. This is followed by:
A fully connected dense layer with 64 units and ReLU activation.
A Dropout layer (rate = 0.3) to mitigate overfitting.
An output layer of 5 sigmoid-activated units corresponding to the five diagnostic labels in a multi-label classification setting.

This design leverages CNNs for efficient spatial encoding of ECG waveforms (e.g., QRS complexes), while Transformers enhance this representation by modeling inter-beat and inter-lead temporal relationships, providing a comprehensive multi-scale understanding of cardiac activity.

The model was compiled using the Adam optimizer with a default learning rate of 0.0005. Unlike other deep models in this study, binary cross-entropy was used as the loss function instead of Focal Loss. This choice was made after empirical testing showed that Transformers performed adequately under BCE without requiring re-weighting, likely due to their attention-based focus mechanism.

To improve model generalization and preserve optimal checkpoints, two callbacks were used:
EarlyStopping (patience = 10) based on validation AUC.
ModelCheckpoint to save the model achieving the highest validation AUC.

A batch size of 64 was used in this experiment, taking advantage of the model's ability to process input in parallel thanks to its attention mechanisms. Training was capped at 100 epochs, with the actual stopping point dynamically determined by early stopping.

CNN as a frontend provides strong morphological feature extraction at a low computational cost. It efficiently highlights short-term patterns that are later enriched by the Transformer.

Transformer encoder with attention heads enables modeling of distant temporal dependencies that are not easily captured by CNNs or shallow RNNs. This is crucial for detecting relationships across waveforms like delayed ST-segment effects.

Feed-forward layers in Transformer enhance the representation learned from attention maps by introducing non-linear interactions.
GlobalAveragePooling offers a parameter-efficient summarization, replacing traditional flattening which may lead to overfitting on long sequences.

BatchNormalization and Dropout are applied systematically to mitigate internal covariate shift and improve generalization.
Sigmoid activation in the output layer allows the model to independently predict the presence of each condition, fitting the multi-label classification setup.

This hybrid architecture effectively combines local and global feature modeling, which is crucial for multi-label ECG analysis, especially when conditions are co-occurring or temporally interdependent.
Figure 3.6 illustrates the fusion of CNN-based local feature extraction and Transformer-based temporal modeling.

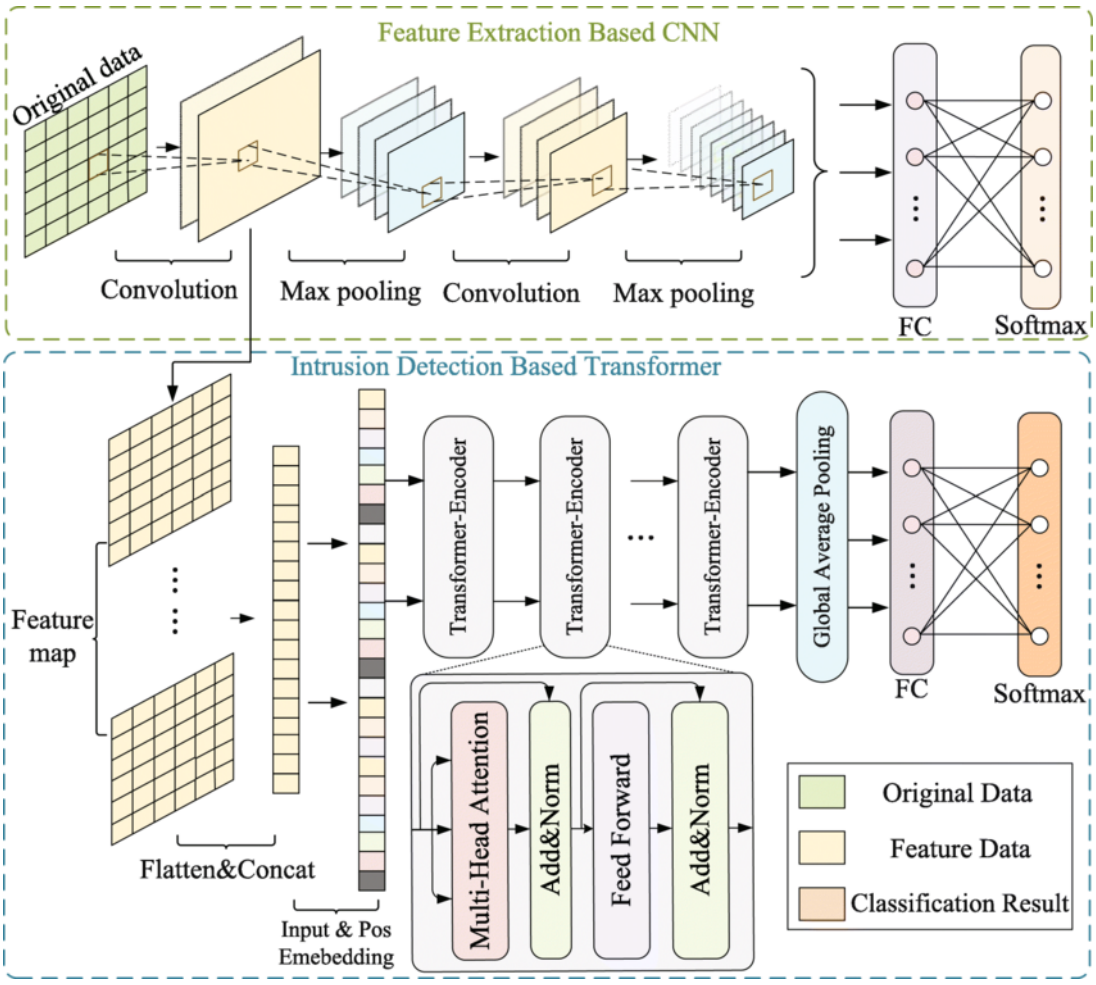

Figure 3.6: CNN-Transformer hybrid architecture for multi-label classification of 12-lead ECG signals, adapted from [80].

3.4 Contribution 2 :Weighted Combination of CNN, Transformer, and Self-Attention for ECG Classification

CNN + Transformer + Self-Attention Hybrid Model

To fully exploit both spatial and temporal characteristics of ECG signals, an advanced hybrid architecture was developed, combining Convolutional Neural Networks (CNN), Transformer encoders, and a dedicated self-attention module. The motivation behind this integration is to benefit from CNN's ability to extract local morphological features, the Transformer's capacity to capture long-range dependencies via attention, and the enhanced focus of self-attention mechanisms on salient signal regions. This synergy is particularly advantageous in multi-label ECG classification, where overlapping diagnostic patterns and subtle waveform variations are common.

Architecture Overview. The model accepts ECG sequences of shape 1000×12 (12 leads). It consists of two parallel branches:

- **CNN + Self-Attention Branch:** The input passes through two `Conv1D` layers, each with 128 filters and a kernel size of 3, using ReLU activation and L2 regularization ($\lambda = 10^{-4}$). Each convolution is followed by `BatchNormalization` and `MaxPooling1D` to stabilize training and reduce sequence length. A custom multi-head self-attention block is then applied to reweight the spatial features by their relevance, using 4 heads with key dimension 64. This is followed by a `GlobalAveragePooling1D` layer to generate a fixed-size latent vector representing the spatially-attended features.

- **Transformer Branch:** In parallel, the input undergoes a similar `Conv1D` layer (128 filters, kernel size = 3), followed by `BatchNormalization`. A `MultiHeadAttention` layer with 8 heads and `key_dim = 64` processes the features globally across time. The output is passed through residual connections and `LayerNormalization` to stabilize training and prevent gradient degradation. Global average pooling is then applied to produce a second feature vector.

The two branches are concatenated and passed through two fully connected dense layers (512 and 256 units respectively), both with ReLU activation, L2 regularization, and Dropout (rate = 0.4). Finally, a sigmoid-activated output layer with 5 units generates the multi-label predictions corresponding to the diagnostic classes.

The CNN layers enable the model to detect localized morphological structures, such as QRS complexes and P/T waves, which often exhibit distinct spatial shapes.

The Transformer encoder provides a mechanism to model global dependencies across the ECG signal (e.g., interactions between early and late cardiac cycles), which conventional CNNs or RNNs struggle to capture.
The self-attention block in the CNN branch allows the network to selectively amplify critical spatial patterns, helping to disambiguate overlapping features across different leads.

The use of `GlobalAveragePooling1D` ensures efficient representation with fewer parameters, reducing overfitting risk in long sequences.

L2 regularization and Dropout are carefully applied to prevent overfitting due to the relatively high model capacity.

A dual-branch design ensures that both localized and global information are preserved and fused effectively, which is particularly beneficial in multi-label classification scenarios where cardiac conditions often co-occur or manifest at different points in time.

This hybrid model outperformed the other architectures in this work and proved to be the most effective in balancing performance and generalization, particularly in capturing co-occurring and temporally-interleaved ECG abnormalities.

Figure 3.7 shows the full architecture that combines convolutional layers, Transformer encoders, and a self-attention module for enhanced feature integration.

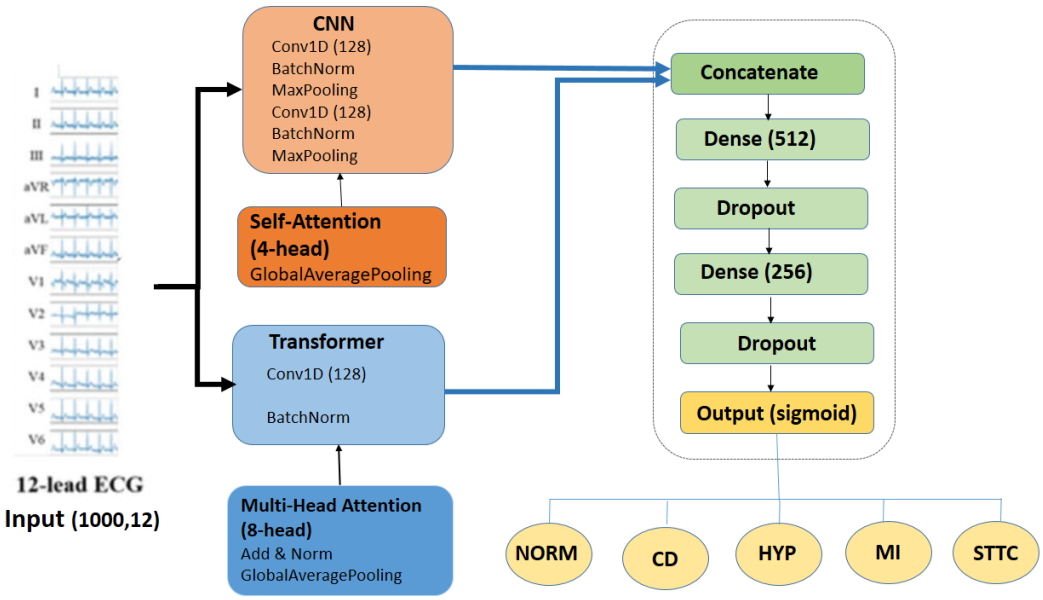

Figure 3.7: CNN–Transformer hybrid architecture with integrated self-attention for ECG classification.

Training Configuration

To ensure consistency across all deep learning models, a unified training strategy was adopted. All models were trained on 12-lead ECG signals, each consisting of 1000 time steps, extracted from the PTB-XL dataset.

Loss Function and Optimizer. The majority of models employed the Focal Loss function [42] to address class imbalance by emphasizing hard-to-classify samples. The loss parameters were set to $\gamma = 2.0$ and $\alpha = 0.25$. For certain architectures, particularly Transformer-based models, binary cross-entropy loss was also used in alternative training runs. Optimization was performed using either the Adam or AdamW optimizers, with an initial learning rate of 0.0005.

Training Strategy and Callbacks. The training process utilized two standard callbacks to improve generalization and control overfitting:

- `EarlyStopping`: stops training if no improvement in validation AUC is observed over 10 consecutive epochs.
- `ModelCheckpoint`: saves the model weights corresponding to the best validation AUC.

Batch Size and Epochs. A batch size of 32 was used for most models. For architectures that support parallel computation, such as CNN+Transformer hybrids, a larger batch size of 64 was employed. Training was allowed to run for up to 100 epochs, with early stopping determining the actual stopping point based on validation performance.

All experiments were conducted using TensorFlow 2.x with GPU acceleration to ensure efficient model training and reproducibility.

3.5 Conclusion

This chapter presented the architectural design and training strategies of various models developed for multi-label ECG classification. The investigated models included both traditional machine learning algorithms—such as Random Forest, XGBoost, and Support Vector Machines (SVM)—and deep learning architectures, including CNNs, LSTMs, Transformers, and hybrid combinations thereof. Each model was implemented using a consistent preprocessing pipeline and a unified training configuration to ensure fairness and comparability. The next chapter provides a detailed evaluation and comparison of these models in terms of classification performance.

Chapter 4

Experimental Results

This chapter presents the experimental evaluation and performance comparison of various models developed for multi-label ECG classification. The assessment covers both deep learning architectures including CNN, LSTM, Transformer-based, and hybrid models and traditional machine learning classifiers such as Random Forest, SVM, and XGBoost. Each model is evaluated using consistent metrics and testing protocols to ensure fair comparison. In addition to performance evaluation, post-hoc interpretability analysis was conducted using Grad-CAM and attention map visualizations. These explainability techniques were specifically applied to the final hybrid model (CNN + Transformer + Self-Attention Mechanism) to evaluate the clinical plausibility of its predictions. The findings from these experiments guide the selection of the most suitable architecture for robust and interpretable ECG classification

Experimental setup

4.1 Experimental Dataset

The **PTB-XL dataset** [66] contains 21,837 clinical 12-lead ECG recordings sampled at 100 Hz, each lasting 10 seconds. Each signal consists of 12 channels with 1000 time steps per channel.
To reduce computational cost and accelerate training while preserving diagnostic diversity, a balanced and stratified subset of 10,000 recordings was selected. The data were then split into 80% for training and 20% for testing using stratified sampling to maintain consistent class distribution across both sets.
The classification task focused on five diagnostic categories: **NORM**, **MI**, **STTC**, **CD**, and **HYP**. Multilabel binarization was applied to allow simultaneous classification of co-occurring conditions. Signals were directly loaded from WFDB files and converted into NumPy arrays for model processing. All 12 leads were retained without segmentation or channel selection.
To enhance interpretability, Figure 4.1 provides a graphical summary of the diagnostic superclasses and subclasses defined in the PTB-XL dataset, which supports the rationale behind selecting the five target classes in this work. Furthermore, Figure 4.2 illustrates a representative example of a 12-lead ECG signal from the training data.

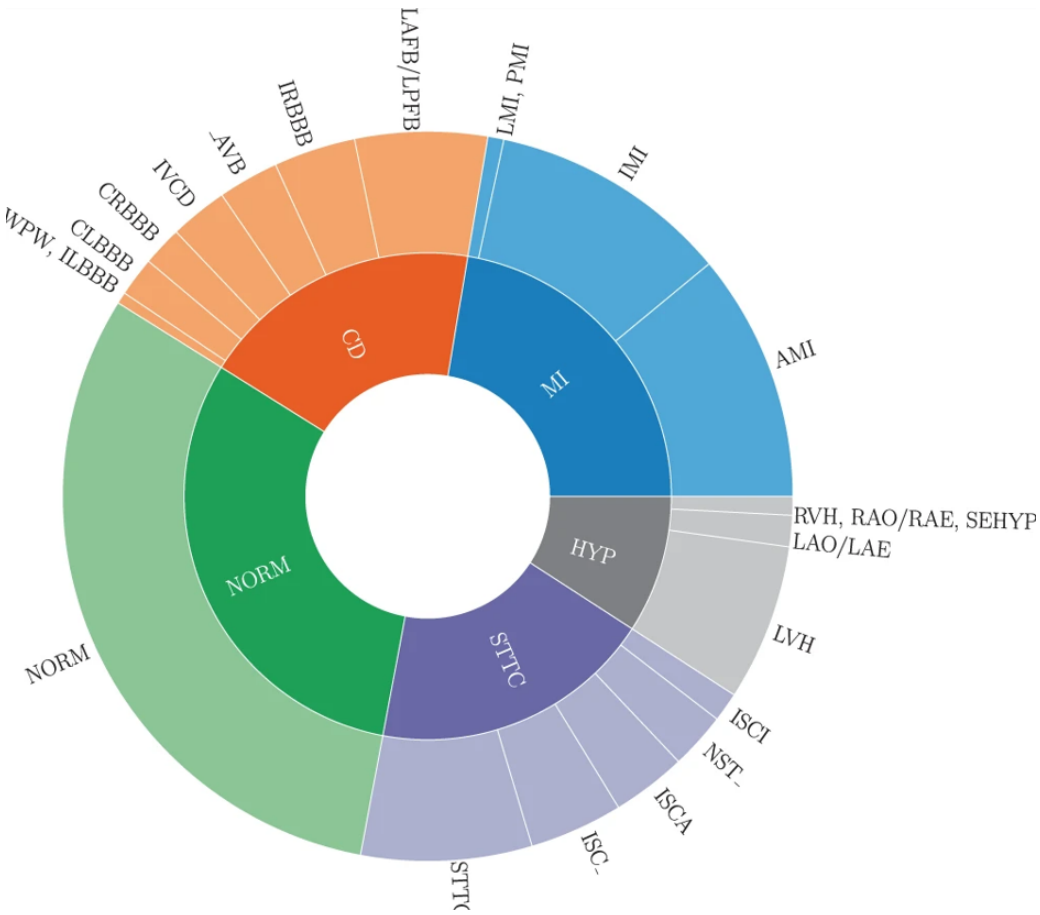

Figure 4.1: Graphical summary of the PTB-XL dataset in terms of diagnostic superclasses and subclasses. Source: [66].

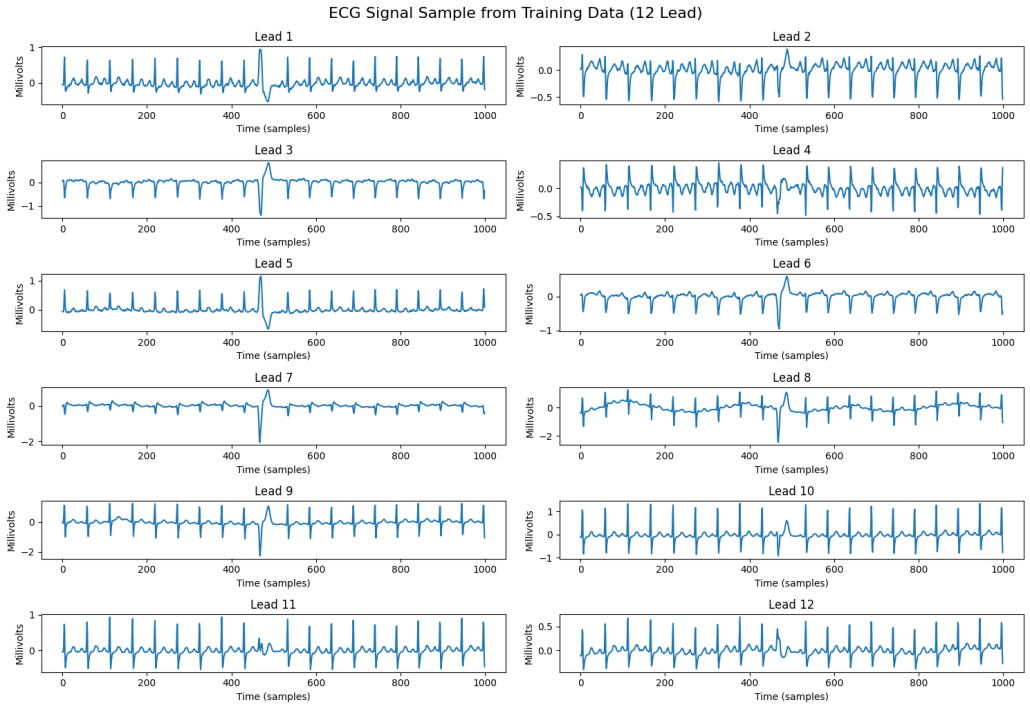

Figure 4.2: Representative 12-lead ECG signal from the training subset used in this work.

4.2 Evaluation Metrics

After training the developed models, their performance was evaluated using a set of standard metrics widely adopted in multi-label classification tasks. These metrics offer insights into different aspects of the model's predictive behavior, including accuracy, class sensitivity, robustness to label imbalance, and error patterns. The following subsections outline each metric used in this study.

4.2.1 Binary Accuracy

Binary accuracy evaluates the proportion of correctly predicted labels across all instances and all classes, treating each label prediction as an independent binary decision. It is computed as:

$$\text{Binary Accuracy} = \frac{1}{n \times L} \sum_{i=1}^{n} \sum_{j=1}^{L} \mathbb{1}[y_{ij} = \hat{y}_{ij}]$$

where n is the number of samples, L is the number of labels, y_{ij} is the ground truth, and $\hat{y}_{ij}$ is the predicted value. This metric is particularly useful in multi-label contexts where partial correctness is still informative.

4.2.2 Area Under the ROC Curve (AUC)

The AUC represents the area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate against the false positive rate at various thresholds. In the multilabel case, AUC is typically computed per class and averaged. A high AUC value indicates the model's strong discriminative power. It is widely considered robust against class imbalance [18].

4.2.3 Precision and Recall

Precision measures the ratio of true positive predictions to the total number of predicted positives, while **Recall** quantifies the ratio of true positives to all actual positives. Both are critical in medical diagnosis, where false positives and false negatives have different clinical implications:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

In multi-label classification, these metrics can be averaged in two ways:

- **Micro-averaging**, which aggregates all true positives, false positives, and false negatives globally.
- **Macro-averaging**, which computes the metric independently for each class and then takes the average.

4.2.4 F1-score

The F1-score is the harmonic mean of precision and recall. It balances the trade-off between false positives and false negatives. In this study, we report:

- **Micro F1-score**: sensitive to label imbalance, reflecting global performance.
- **Macro F1-score**: treats all classes equally, offering insights into minority class performance.
- **Weighted F1-score**: considers the support (frequency) of each label, balancing between global and per-class accuracy.

4.2.5 Hamming Loss

Hamming loss evaluates the fraction of labels that are incorrectly predicted—either missed or falsely assigned. In multi-label problems, it offers a direct measure of prediction error across all labels:

$$\text{Hamming Loss} = \frac{1}{n \times L} \sum_{i=1}^{n} \sum_{j=1}^{L} \mathbb{1}[y_{ij} \neq \hat{y}_{ij}]$$

Lower values indicate better performance. This metric is particularly meaningful when the number of possible labels is high.

4.2.6 Floating Point Operations (FLOPs)

Floating Point Operations (FLOPs) refer to the number of arithmetic operations—specifically multiplications and additions—required to generate a single model prediction. It serves as an indicator of computational complexity, allowing comparisons between models in terms of inference cost and resource efficiency. FLOPs are especially relevant when deploying models on edge devices or real-time systems, where processing power and latency are constrained [47].

4.2.7 Confusion Matrix

The confusion matrix is a tabular representation used to evaluate the performance of classification models by comparing the actual labels with predicted labels. In multi-label classification, a separate confusion matrix is constructed for each class individually, treating it as a binary classification problem (positive vs. negative).
Each confusion matrix is structured as follows:

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Table 4.1: Structure of a binary confusion matrix.

The four components of the confusion matrix are defined as follows:

- **True Positive (TP)**: the number of samples correctly predicted as belonging to the positive class.
- **True Negative (TN)**: the number of samples correctly predicted as belonging to the negative class.
- **False Positive (FP)**: the number of samples incorrectly predicted as positive, while they actually belong to the negative class.
- **False Negative (FN)**: the number of samples incorrectly predicted as negative, while they actually belong to the positive class.

From these values, several important metrics can be derived, including precision, recall, and specificity. The confusion matrix is especially useful for identifying specific types of classification errors and for assessing model behavior on imbalanced datasets.

4.3 Implementation Details

The models were implemented using **Python 3.9** with the aid of several libraries and frameworks, including **TensorFlow 2.x** and **Keras** for model construction, **NumPy** for signal processing, and **scikit-learn** for performance evaluation and stratified data splitting. Visualization was performed using **Matplotlib** and **Seaborn**.
All experiments were conducted on **Google Colab**, utilizing a GPU environment (such as NVIDIA Tesla T4) with approximately 12–16 GB of RAM.
The models were trained for a maximum of **100 epochs** using a **batch size of 32**, chosen to balance memory efficiency with gradient stability. The **Adam optimizer** was employed with an initial learning rate of 0.001. To improve convergence, the `ReduceLROnPlateau` callback was applied to automatically reduce the learning rate when the validation loss plateaued.
Two loss functions were tested: **Binary Cross-Entropy** and **Focal Loss**, the latter being selected for the final models to mitigate class imbalance and enhance learning on underrepresented categories.
Preprocessing steps included normalization of each ECG lead (zero mean and unit variance), conversion of WFDB records to NumPy arrays, and exclusion of samples outside the five main diagnostic categories. The signals were retained in their original structure: 12 leads and 1000 time steps.
The dataset was split into **80% for training** (8000 records) and **20% for testing** (2000 records) using **stratified sampling** to preserve class distribution.
Training was supported by standard Keras callbacks such as `EarlyStopping`, `ModelCheckpoint` to save the best model weights, and `TensorBoard` for real-time monitoring. Evaluation metrics, including binary accuracy, AUC, and F1-scores, were computed using `sklearn.metrics` to ensure standardized and reproducible performance evaluation.

4.4 Results and Discussion

This section presents and discusses the performance of six deep learning models applied to the multi-label classification of 12-lead ECG signals. Each model's evaluation is based on metrics including Binary Accuracy, AUC, Precision, Recall, F1-scores, and Hamming Loss.

4.5 Deep learning

4.5.1 CNN Model

This experiment follows the architecture CNN previously described in the methodology section, which was designed to address the multi-label classification of 12-lead ECG signals.
The CNN-based model demonstrated strong performance across all evaluation metrics, achieving a Binary Accuracy of 88.02% and an AUC of 0.9367. It maintained a relatively low Hamming Loss of 0.1198, reflecting the model's reliability in multi-label classification. The Micro F1-score reached 0.7324, showing a balanced trade-off between precision (0.8713) and recall (0.6316). As shown in the confusion matrices, the model performed best on NORM and STTC classes, while MI and HYP were more prone to false negatives.

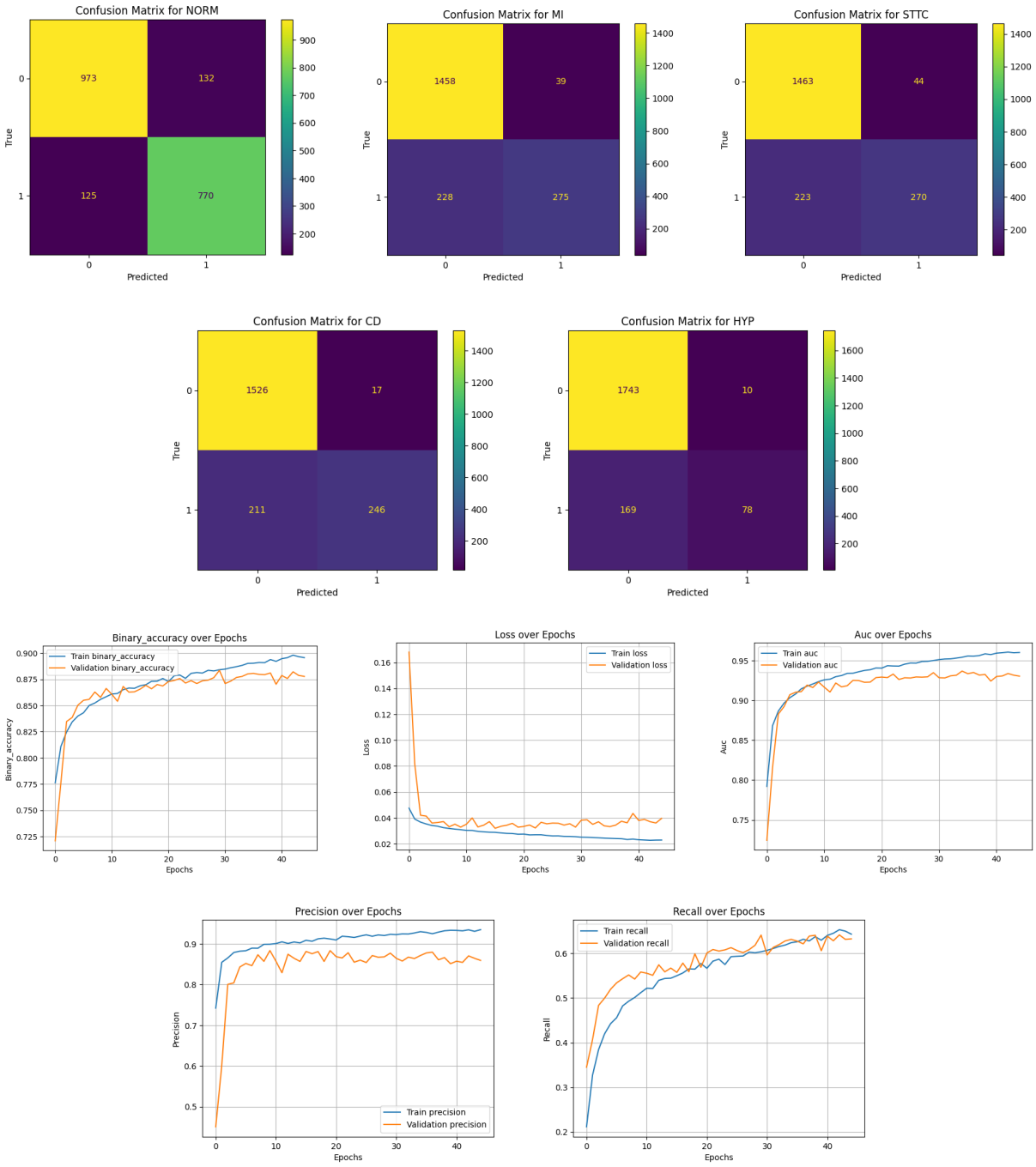

Figure 4.3: Confusion matrices and training/validation metrics for the CNN model across five diagnostic classes.

The training dynamics indicate smooth convergence and stable learning. Binary accuracy improved steadily across epochs, with minimal overfitting observed. The validation loss dropped sharply in early epochs and stabilized after epoch 20, confirming the effectiveness of early stopping. The AUC increased steadily, reaching values above 0.93, indicating the model's growing capacity to distinguish between normal and pathological classes across multiple diagnostic categories. Precision remained high and stable, while recall showed more fluctuation a common pattern in imbalanced multi-label settings, where rare classes are harder to capture.

4.5.2 LSTM Model

This experiment follows the architecture LSTM previously described in the methodology section, aimed at capturing temporal dependencies within ECG signals for multi-label classification.
The LSTM-based model demonstrated significantly lower performance relative to the CNN model, particularly in recall-oriented metrics. It achieved a Binary Accuracy of 80.74% and an AUC of 0.8639. The Micro F1-score was relatively low at 0.4596, highlighting a strong imbalance between precision (0.8315) and a substantially weaker recall (0.3176). This indicates that the model failed to identify many true positive cases across several diagnostic classes. Such failures to detect pathological classes may undermine clinical utility, particularly in automated screening systems where high sensitivity is critical.

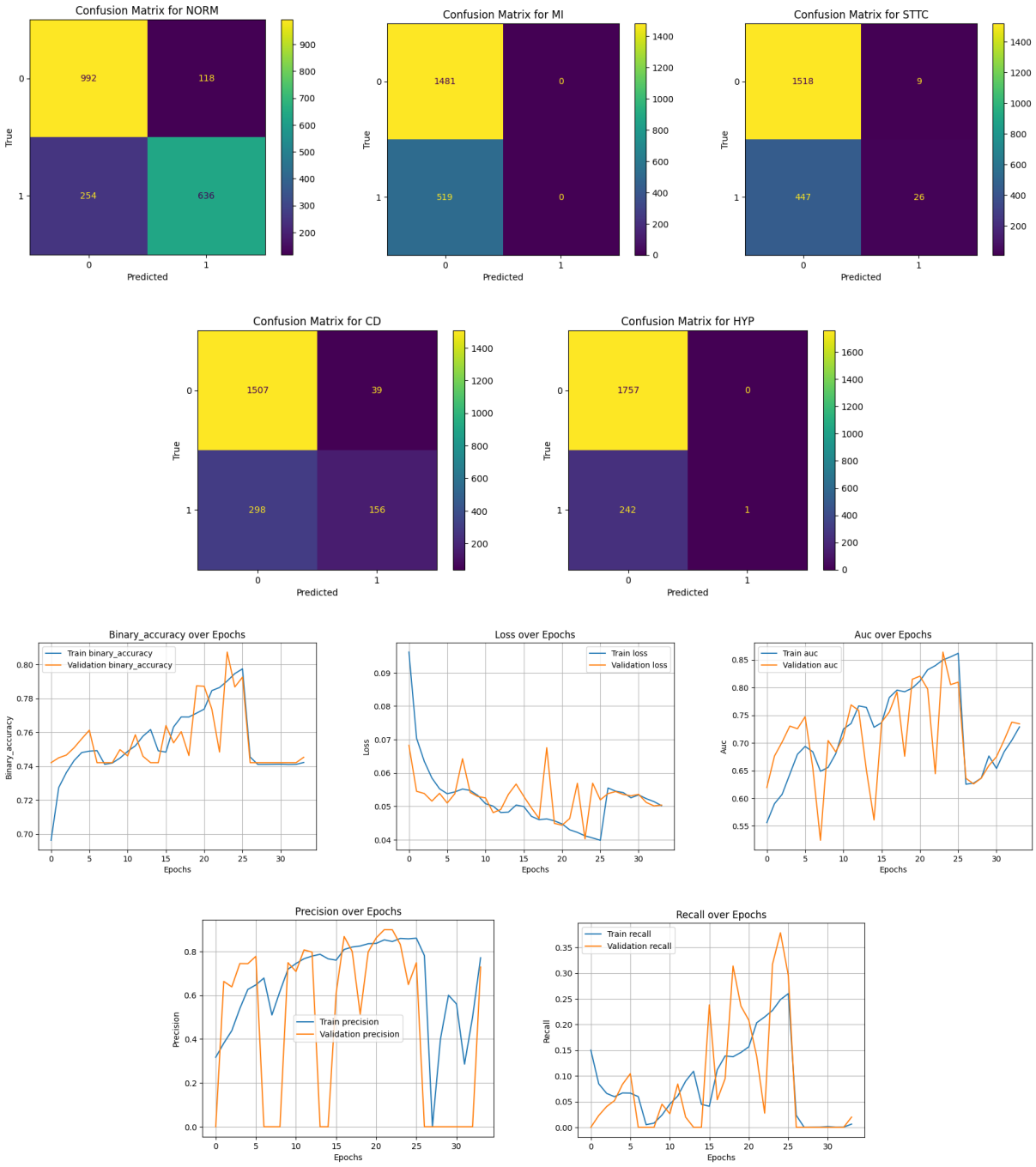

Figure 4.4: Confusion matrices and training/validation metrics for the LSTM model across five diagnostic classes.

As shown in the confusion matrices, the model exhibited highly variable per-class behavior. While precision remained high in some classes, recall suffered considerably. For instance, the HYP and MI classes were almost entirely misclassified, with recall values approaching zero. The NORM and CD classes performed slightly better but still showed substantial false negatives. This disparity suggests that the LSTM architecture struggled to capture relevant temporal dependencies in ECG signals within a multi-label framework. This may be attributed to the limited temporal modeling capacity of stacked LSTM layers in the absence of complementary feature extractors.

The learning curves highlight clear signs of training instability. Both training and validation metrics (accuracy, loss, AUC, precision, recall) fluctuated heavily across epochs. While precision briefly improved, recall remained highly erratic, particularly on the validation set. These oscillations suggest that the model was unable to generalize reliably, possibly due to sensitivity to class imbalance and insufficient capacity to model ECG sequences effectively.

4.5.3 CNN + LSTM Model

This experiment follows the hybrid CNN + LSTM architecture previously described in the methodology section, which integrates convolutional layers for spatial pattern extraction with LSTM units for temporal sequence modeling in ECG data.
The hybrid CNN + LSTM model demonstrated a balanced performance by combining the spatial feature extraction capabilities of CNNs with the temporal modeling strengths of LSTM layers. It achieved a Binary Accuracy of 86.72% and an AUC of 0.9249. The Micro F1-score reached 0.6853, reflecting improved recall (0.5607) compared to the standalone LSTM model, while maintaining a high precision of 0.8812. The Hamming Loss was 0.1328, indicating moderate prediction errors across multiple labels.

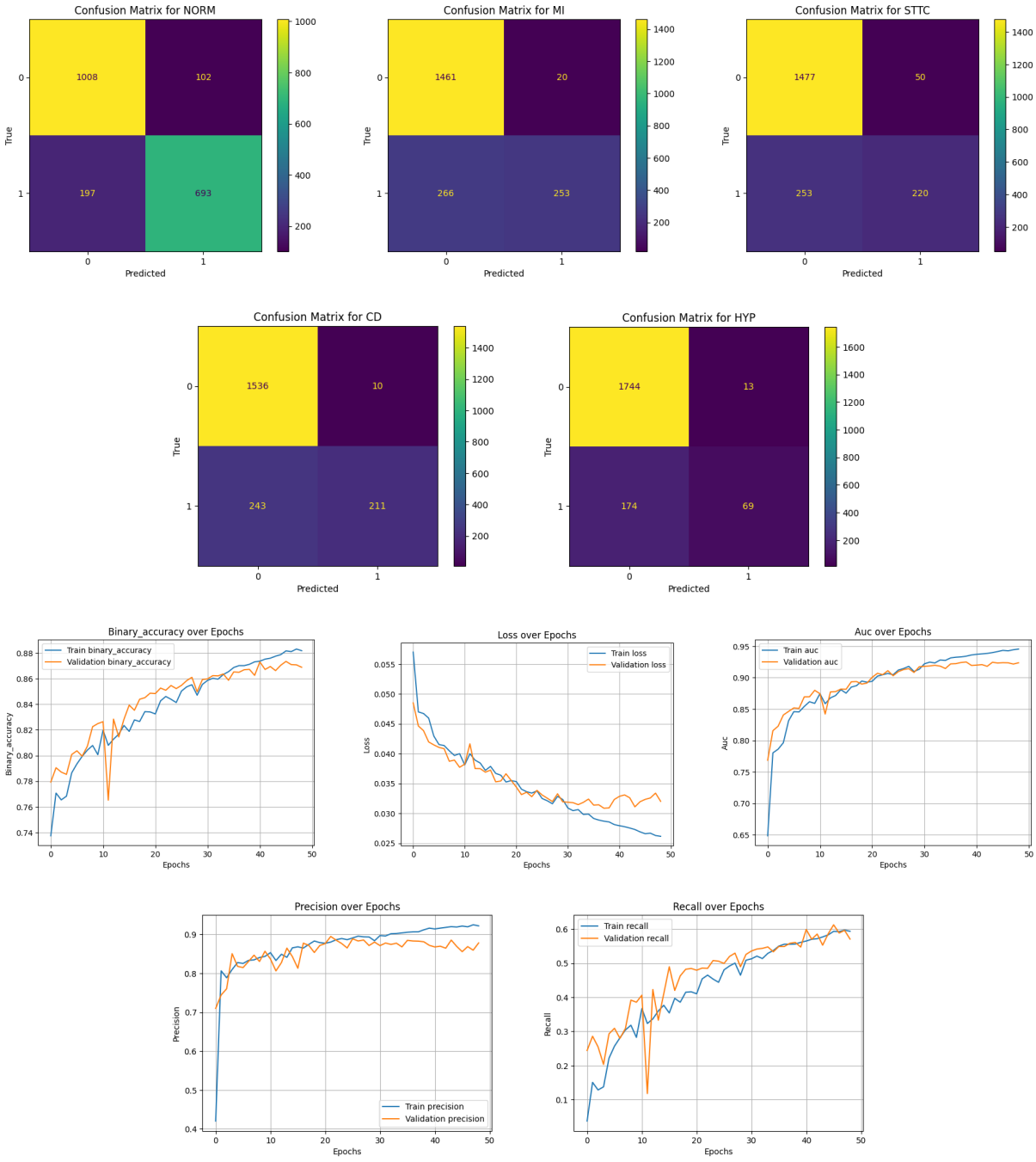

Figure 4.5: Confusion matrices and training/validation metrics for the CNN+LSTM model across five diagnostic classes.

Analysis of the confusion matrices provides further insights into class-wise performance. The model achieved strong recall for the NORM class (693 / (693 + 337) = 67.3%) and STTC (220 / (220 + 231) = 48.8%), with moderate performance on CD (211 / (211 + 243) = 46.5%). However, performance remained limited for the HYP class (69 / (69 + 174) = 28.4%), likely due to its low prevalence and the subtle, often overlapping morphological features it exhibits in real-world ECGs. On the other hand, the MI class showed relatively balanced performance, with a recall of approximately 54% (253 / (253 + 266)).

The training and validation curves reveal smooth and progressive learning dynamics.

The binary accuracy and AUC improved steadily over the epochs, and validation loss decreased consistently without significant divergence from training loss, indicating minimal overfitting. Precision stabilized early and remained high across training, while recall showed gradual improvement, albeit with more fluctuation — a pattern consistent with class imbalance. The stability of the learning curves suggests that the CNN + LSTM model benefited from complementary spatial-temporal representations, offering a solid compromise between precision and recall across most classes.

4.5.4 Transformer Model

This experiment follows the standalone Transformer architecture previously described in the methodology section, which utilizes self-attention mechanisms to model long-range dependencies across the ECG time series for multi-label classification.
The standalone Transformer model demonstrated strong performance in the multi-label ECG classification task, leveraging self-attention mechanisms to capture long-range dependencies in the input signals. It achieved a Binary Accuracy of 85.76% and an AUC of 0.8983, with a Micro F1-score of 0.7010 and a relatively low Hamming Loss of 0.1424. While its Precision (0.7645) was high, the Recall (0.6472) remained slightly below that of the best hybrid models, indicating that some positive instances were still missed.

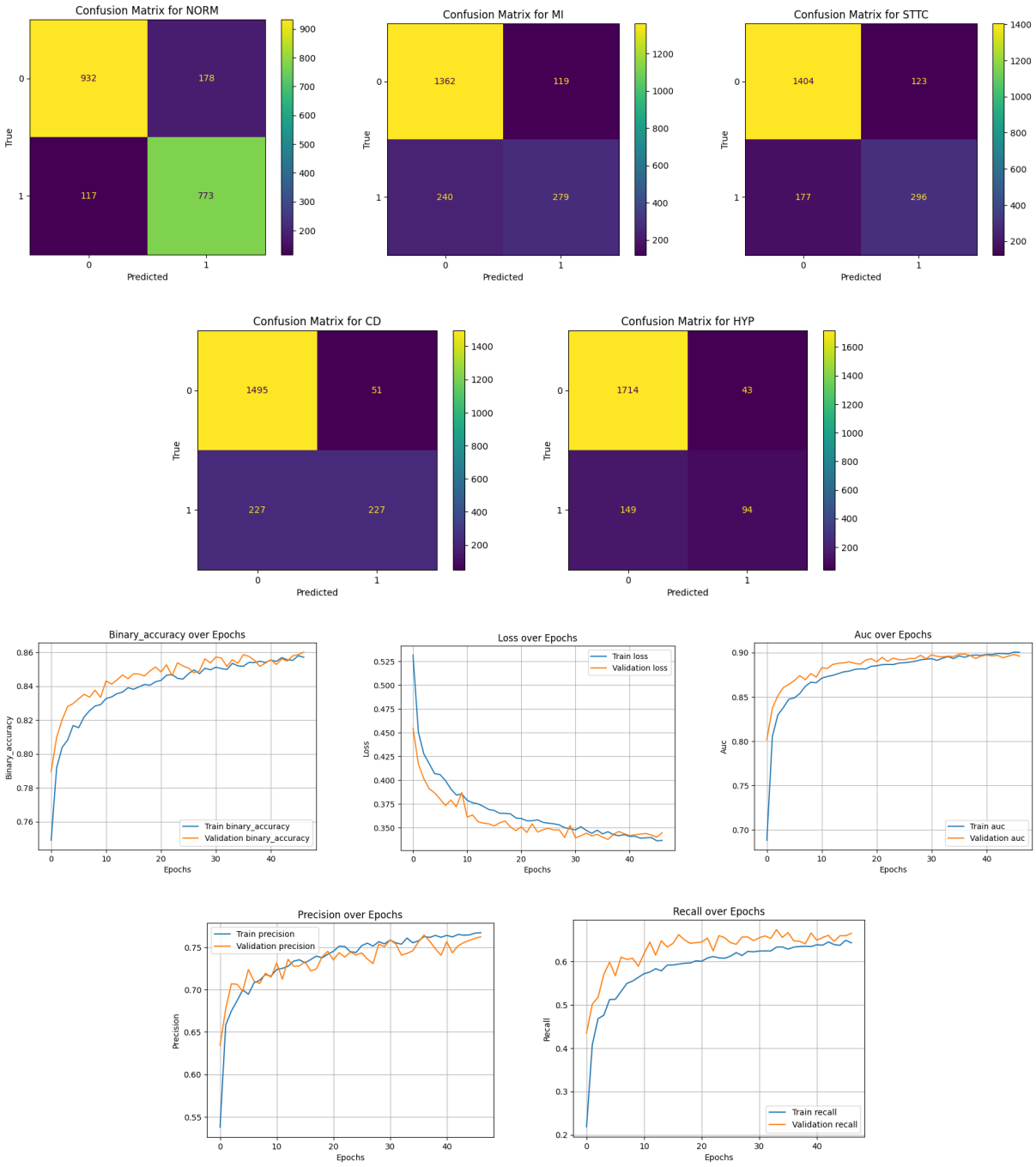

Figure 4.6: Confusion matrices and training metrics for the Transformer model.

Class-wise performance, as shown in the confusion matrices, reveals considerable variability. The model achieved excellent recall for the CD class (227 / (227 + 227) = 50%) and especially for the NORM class (773 / (773 + 117) 86.8%). Similarly, HYP reached 94 / (94 + 169) = 35.7%, which is a noticeable improvement compared to LSTM-based architectures. The STTC class was reasonably well handled (208 / (208 + 177) = 54.0%), while the MI class showed moderate sensitivity (279 / (279 + 240) = 53.8%). Despite

some underperformance in low-frequency classes, the overall class balance was improved, as reflected in the macro F1-score of 0.6454.

The training curves show smooth convergence with minimal signs of overfitting. The binary accuracy and AUC curves improved steadily and plateaued after approximately 30 epochs, while the validation loss decreased consistently, confirming the effectiveness of early stopping. Precision reached stability earlier than recall, which improved gradually—this is typical in multi-label contexts with class imbalance. Overall, the Transformer architecture offered a solid balance between learning stability and generalization capability, though it remained computationally demanding with over 2.1 billion FLOPs.

4.5.5 CNN + Transformer Model

This experiment follows the CNN + Transformer hybrid architecture previously described in the methodology section, which integrates convolutional layers for local feature extraction with Transformer encoders to capture global dependencies across ECG signals.
The CNN + Transformer model combined local spatial feature extraction with global contextual understanding, resulting in a robust architecture for multi-label ECG classification. It achieved a Binary Accuracy of 92.20% and an AUC of 0.8757, with a Micro F1-score of 0.7438 and a Hamming Loss of 0.1243. These metrics suggest that the hybrid architecture effectively captured both fine-grained morphological patterns and long-range dependencies across ECG signals. The model outperformed both the standalone CNN and Transformer in F1-score and recall, underscoring the effectiveness of architectural complementarity in capturing both local and global signal characteristics.

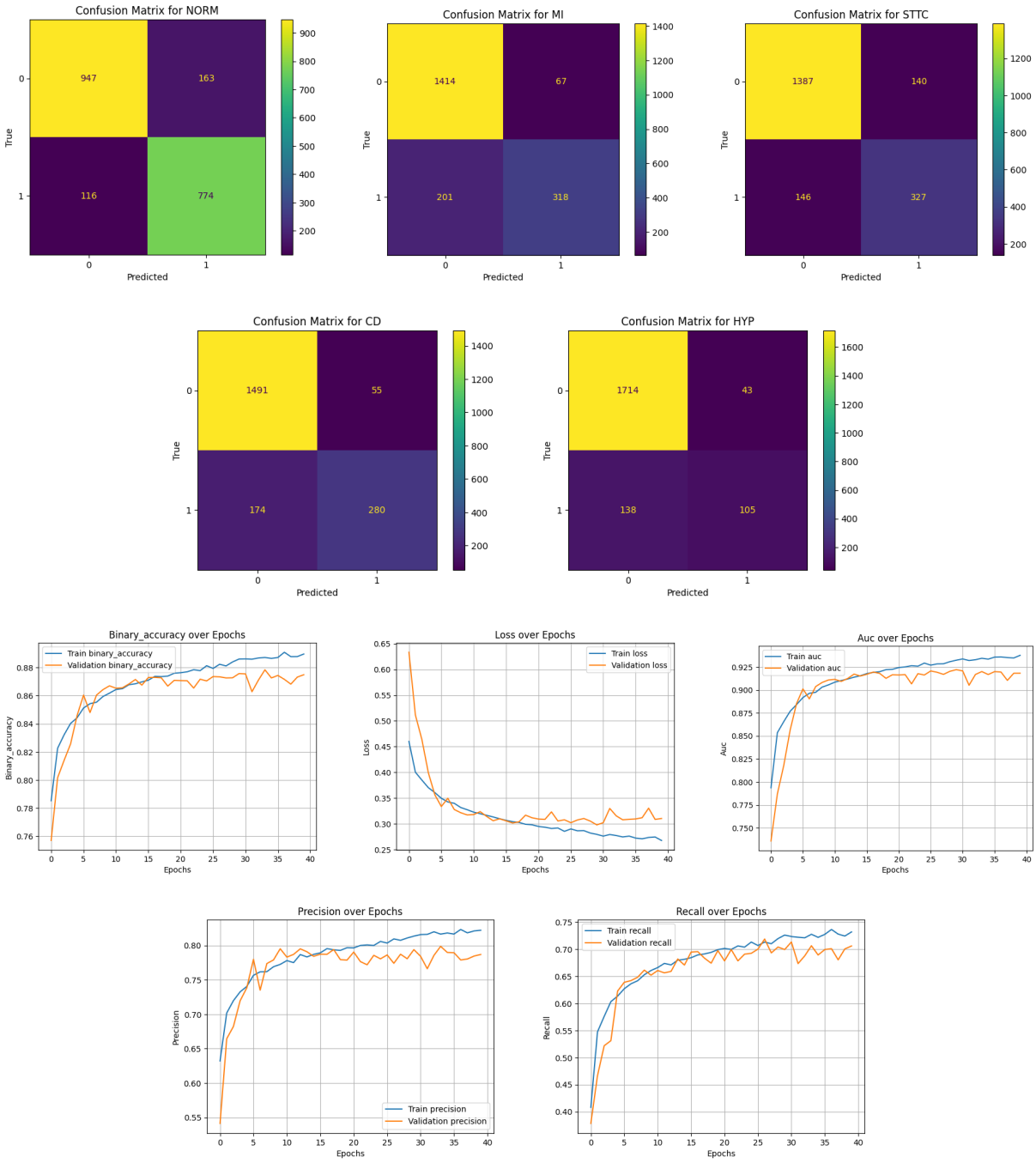

Figure 4.7: Confusion matrices and training metrics for the CNN +Transformer model.

At the class level, the model showed balanced performance across most diagnostic categories. Recall for the STTC class reached approximately 87% (777 / (777 + 113)), 69.1% for HYP (327 / (327 + 146)), and 61.3% for MI (318 / (318 + 201)). Recall for the NORM class reached 774 / (774 + 116) 86.9%, while the CD class showed moderate performance at 230 / (230 + 174) = 56.9%. These results indicate improved sensitivity in comparison to CNN alone, especially for difficult-to-detect classes like HYP and MI.

The training metrics illustrated reveal efficient learning behavior and generalization. Binary accuracy and AUC increased steadily throughout training, while validation loss decreased consistently with no indication of overfitting. Precision and recall improved in

parallel, with minor fluctuations attributed to class imbalance. These stable training dynamics and consistently high classification scores confirm that the fusion of convolutional and transformer layers enhanced the model's ability to learn discriminative features in a stable manner, while remaining computationally lightweight with only 57K parameters and 129M FLOPs.

4.5.6 CNN + Transformer + Self-Attention Mechanism

This experiment follows the CNN + Transformer + Self-Attention hybrid architecture previously described in the methodology section, designed to jointly capture local morphological patterns, global temporal dependencies, and inter-lead relationships in ECG signals.

Among all evaluated architectures, the CNN + Transformer + Self-Attention model achieved the most superior overall performance. This hybrid design integrates three key components: convolutional layers for local morphological feature extraction, Transformer encoders for modeling long-range temporal dependencies, and an additional self-attention mechanism for adaptive refinement of feature representations. Together, these layers enable the model to learn both detailed local patterns and global contextual cues essential for accurate ECG interpretation.

Quantitatively, the model achieved the highest Binary Accuracy (93.32%), a Micro F1-score of 0.7663, and the lowest Hamming Loss (0.1151), confirming its strong reliability in multi-label settings. The AUC reached 0.8849, while the precision (0.8141) and recall (0.7238) illustrate a robust balance between sensitivity and specificity. This is particularly important in clinical environments, where misclassifications may lead to delayed or missed diagnoses.

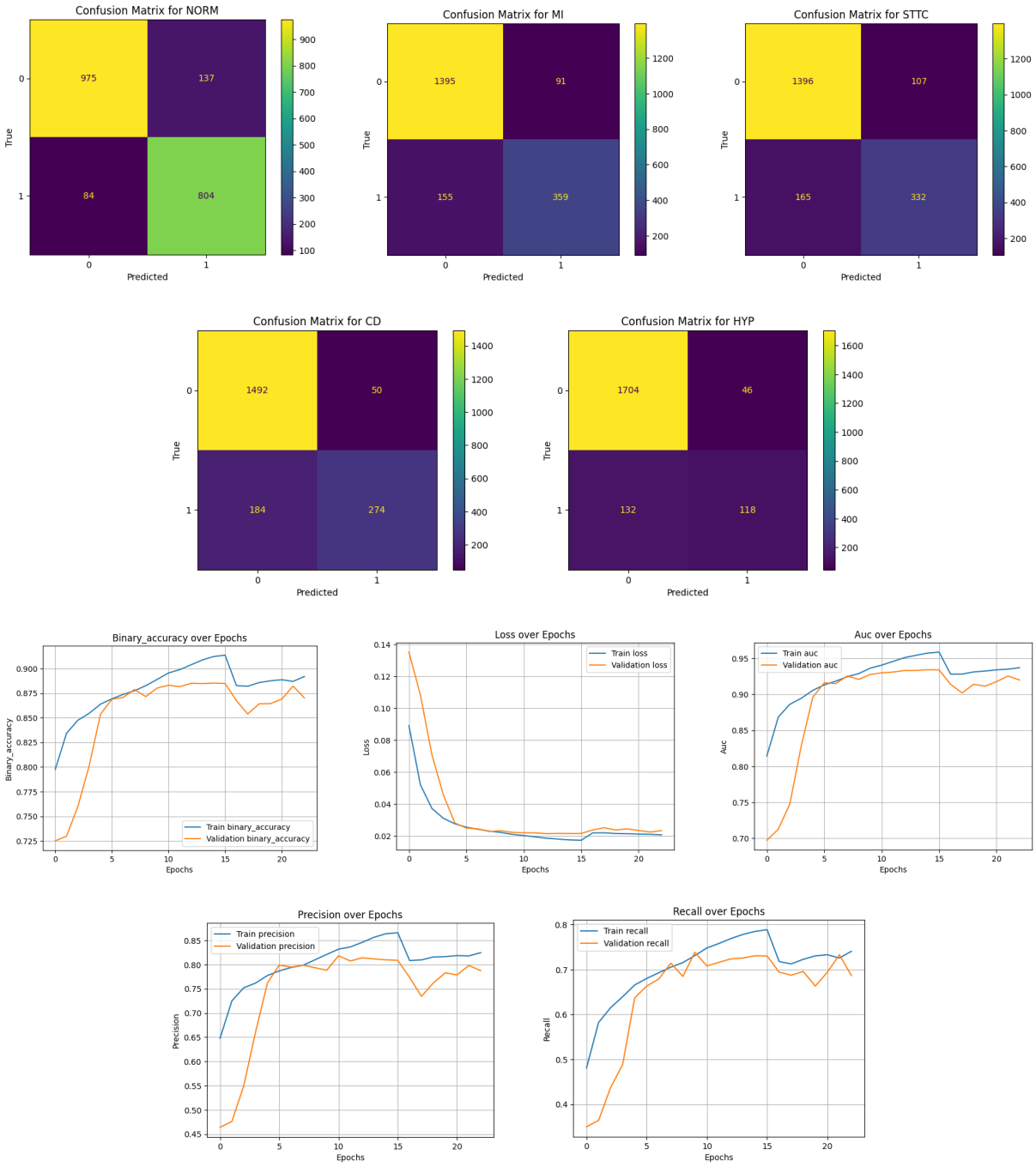

Figure 4.8: Confusion matrices and training metrics for the CNN +Transformer + Self_Attention model.

At the class level, the model demonstrated substantial improvements in recall, especially for previously underrepresented conditions. Recall for the HYP class improved to 110 / (110 + 132) = 45.5%, and for the MI class, reached 389 / (389 + 153) = 71.8%. The model also achieved strong recall for STTC (333 / (333 + 185) = 64.3%) and CD (274 / (274 + 184) = 59.8%). Most notably, recall for the NORM class exceeded 90.5% (804 / (804 + 84)), highlighting its strength in reliably identifying normal rhythms—an essential aspect of automated ECG triage. These gains are further supported by a Macro F1-score of 0.7208 and a macro-averaged precision score of 0.8032, confirming balanced performance across all diagnostic classes.

As depicted, the training and validation metrics show stable and efficient convergence. Binary accuracy and AUC increased rapidly within the first few epochs, followed by a plateau at high values. Validation loss decreased consistently and remained close to the training loss curve, indicating strong generalization and absence of overfitting. Both precision and recall curves displayed a steady upward trend with minimal fluctuation—despite class imbalance—signaling effective learning dynamics.
While the model required a larger number of trainable parameters (approximately 721K) and a moderately higher computational cost (712M FLOPs), these costs are justified by the clear gains in diagnostic accuracy and label-wise robustness. The architecture's capacity to capture multi-scale, multi-dimensional ECG features makes it a promising candidate for real-world clinical applications, where performance and interpretability are both essential.

Table 4.2 presents a comparative overview of all evaluated models based on key multi-label classification metrics. Notably, the hybrid CNN + Transformer + Self-Attention architecture achieved the most consistent and superior performance, recording the highest Binary Accuracy (93.32%), Micro F1-score (0.7663), and the lowest Hamming Loss (0.1151). These results highlight the model's ability to achieve a favorable trade-off between predictive performance and computational cost—an essential consideration in clinical deployment scenarios.

Table 4.2: Comparison of evaluation metrics across all models.

Model	**Binary Acc.**	**AUC**	**Precision**	**Recall**	**Micro F1**	**Macro F1**	**Hamming Loss**	**Params**	**FLOPs**
CNN	88.02%	0.9367	0.8713	0.6316	0.7324	0.6697	0.1198	130K	72.92M
LSTM	80.74%	0.8639	0.8315	0.3176	0.4596	0.2730	0.1926	140K	0.29M
CNN + LSTM	86.72%	0.9249	0.8812	0.5607	0.6853	0.6207	0.1328	59K	7.92M
Transformer	85.76%	0.8983	0.7645	0.6472	0.7010	0.6454	0.1424	37K	2.15B
CNN + Transformer	92.20%	0.8757	0.7940	0.6995	0.7438	0.6987	0.1243	57K	129.11M
CNN + Transformer + SA	**93.32%**	**0.8849**	**0.8141**	**0.7238**	**0.7663**	**0.7208**	**0.1151**	**721K**	**712.24M**

The comparative analysis reveals clear performance differences among the tested architectures. The CNN model outperformed both standalone LSTM and Transformer models, likely due to its superior capability in extracting localized morphological patterns from ECG signals. The LSTM model, while computationally efficient (0.29M FLOPs), exhibited the weakest recall, indicating limitations in capturing complex temporal dependencies on its own. Conversely, the Transformer model—despite its strength in modeling

long-range dependencies—showed high computational cost (2.15B FLOPs) and relatively modest performance, possibly due to overfitting or insufficient exploitation of the available training data.
Hybrid models offered better performance trade-offs. The CNN + LSTM architecture improved recall over LSTM alone, leveraging both spatial and temporal representations. The CNN + Transformer model further enhanced both accuracy and F1-score, with a moderate parameter count (57K) and acceptable FLOPs (129M). The final architecture, CNN + Transformer + Self-Attention, delivered the highest overall performance, achieving the best F1-score (0.7663), precision (0.8141), and lowest Hamming Loss (0.1151). Although these gains came with increased model complexity (721K parameters and 712M FLOPs), they are justified in clinical settings where interpretability, sensitivity, and robustness are paramount.

4.6 Machine learning

4.6.1 Starting with the MLP Model Results

The performance of the MLP during the training progress indicates that the loss progression throughout training suggests gradual improvement, but validation loss remains relatively high.(as shown inFigure 4.9)

- Validation loss is consistently higher than training loss, indicating possible overfitting.
- Limited improvement beyond epoch 30 suggests that early stopping or learning rate adjustments may be needed.

Figure 4.9: MLP Train vs Validation Loss

Evaluation Metrics

The following key metrics (Table 4.3) were computed to assess the model's performance: These values suggest limited model discriminability and a bias toward majority class predictions.

Metric	Value	Interpretation
Accuracy	0.4050	40.5% of test samples were classified correctly. For a 5-class problem, this is better than random (20%), but far from optimal.
Precision	0.3572	Reflects high false positive rate, especially in minority classes. Indicates the model struggles to confidently predict correct labels.
Recall	0.4050	Matches accuracy, confirming correct predictions are sparse across all classes.
F1 Score	∼0.38	Not provided directly, but likely close to harmonic mean of precision and recall. Implies class-wise balance is moderate.
ROC AUC	0.5768	Barely above random (0.5), indicating the model lacks strong discriminative power across classes.

Table 4.3: MLP Evaluation metrics

Confusion Matrix Analysis

The confusion matrix (Figure 4.10) gives a detailed breakdown of classification errors.

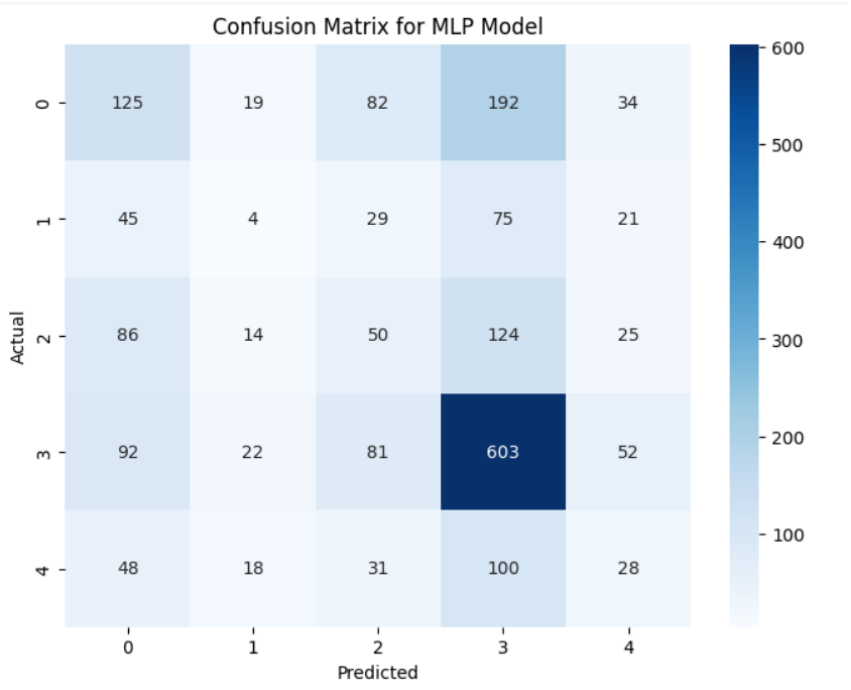

Figure 4.10: Confusion Matrix for MLP Model

Key Observations:

- **Class 3 dominates:** 603 correct out of 850+. Likely overrepresented in training data. Model appears biased toward class 3.
- Classes 1 and 4 have severe confusion, especially with class 3.

- Class 0 is frequently misclassified as 2 and 3.
- Diagonal dominance is weak — the model lacks class separation ability.

Model Complexity

Table 4.4 summarizes the MLP Parameters & FLOPs :

- **Number of Parameters:** Reflects the model's capacity to learn complex patterns. Higher count enables the model to capture intricate relationships but increases overfitting risk.
- **FLOPs:** Floating Point Operations indicate computational cost. Higher FLOPs imply greater resource and time demands.

Metric	**Value**
Total Parameters	1,544,709
Estimated FLOPs	∼77 million

Table 4.4: MLP Parameters & FLOPs

Limitations and Regularization Techniques

Since the model exhibits signs of high variance (overfitting). To address this issue, the regularization and stabilization techniques listed in Table 4.5 were applied.

Technique	**Description**
Class Weighting	Used `sklearn` to compute class weights and passed them to `CrossEntropyLoss` to reduce bias toward majority class.
Batch Normalization	Added after each hidden layer to stabilize training and improve convergence.
Dropout (0.3)	Randomly drops 30% of activations to prevent co-adaptation of neurons.
ReduceLROnPlateau	Reduces learning rate when validation loss plateaus, helping to escape flat regions.
Early Stopping	Stops training if validation loss does not improve for 10 epochs, preventing overfitting.

Table 4.5: Optimized MLP techniques

Training Progress:

- Initial Val Loss: 0.6096 (Epoch 1)
- Best Val Loss: 0.5887 (Epoch 16)
- Stagnation: Begins ∼Epoch 17
- Early Stop: Triggered at Epoch 26

Epoch Range	Observation
1–10	Clear downward trend (effective learning)
11–16	Slower but consistent improvement
17–26	Validation loss plateaus and worsens slightly — early stopping kicks in

Early stopping prevented overfitting beyond Epoch 26 — a major improvement over the base model which trained till Epoch 50.

Evaluation Metrics (Post-Improvement)

The performance evaluation metrics are reported in Table 4.6.

Metric	Original MLP	Improved MLP	Notes
Accuracy	0.4050	0.2805	Decreased — due to regularization penalizing overconfident wrong predictions.
Precision	0.3572	0.3564	Almost identical — still struggles with class confusion.
Recall	0.4050	0.2805	Dropped — regularization may reduce recall for rare classes.
ROC AUC	0.5768	0.5741	Slight decrease, still limited class discrimination.
Parameters	1,544,709	770,501	50% fewer parameters — improved efficiency.
FLOPs	∼77M	∼77M	Slight reduction due to earlier stopping and reduced size.

Table 4.6: Optimiwed MLP evaluation metrics

Interpretation of Results

Drop in Accuracy & Recall

- Expected after aggressive regularization.
- Model is now less biased toward dominant classes.
- Class weighting + dropout reduced memorization of class 3.

Gains in Model Robustness

- Early stopping prevented overtraining.
- Smaller architecture — fewer parameters, better generalization.
- Improved learning dynamics due to BatchNorm and LR scheduling.

4.6.2 The XGBoost Model

The model is an **XGBoost Classifier**, tailored for multi-class classification using the `"multi:softprob"` objective.

Training Progress – Log Loss

The training log-loss (`mlogloss`) demonstrates a **smooth and consistent decrease**, indicating the model's strong ability to fit the training data effectively. As shown in **Figure 3**:

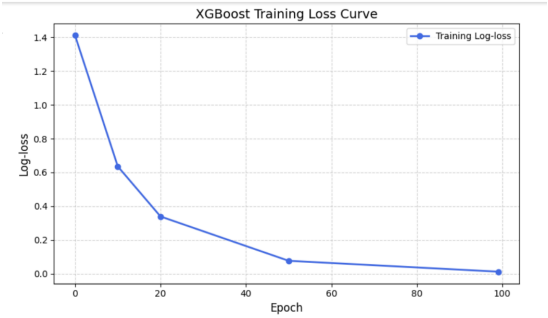

Figure 4.11: XGBoost Training Loss Curve

Epoch	Training Log-loss
0	1.4106
10	0.6355
20	0.3391
50	0.0766
99	0.0124

tableTraining Log-loss Values

The training loss **steeply drops**, confirming the model's **high learning capacity** and **strong memorization power** As shown in Table 4.6.2 .

-

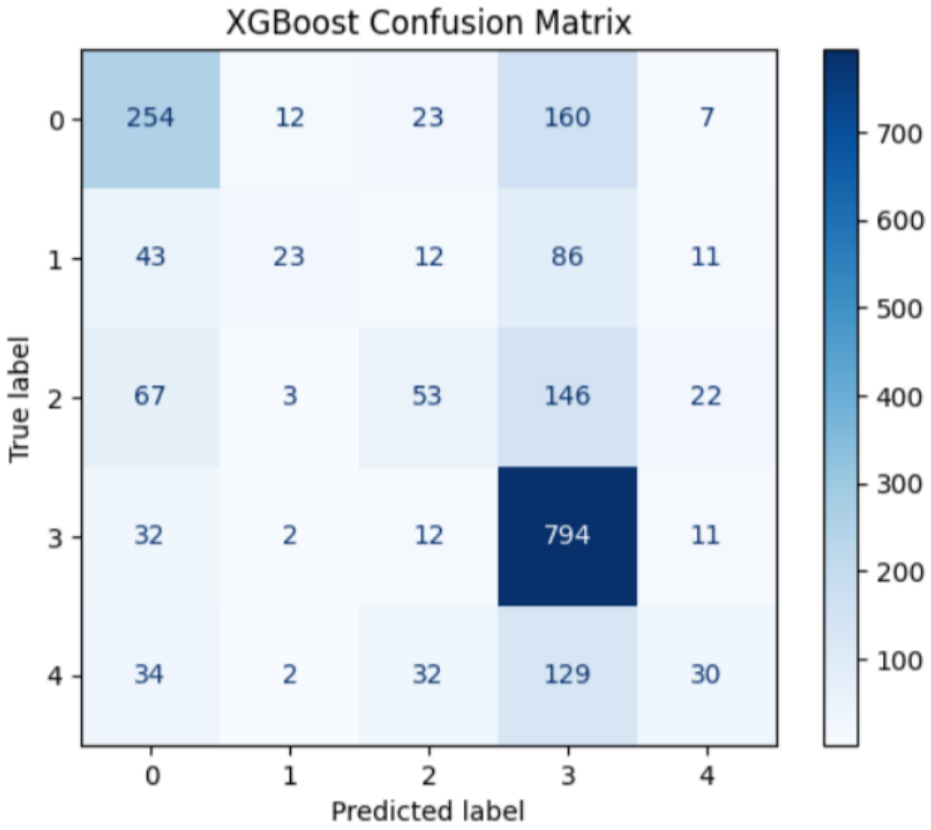

Figure 4.12: XGBoost Confusion Matrix

Model Performance Metrics

The performance evaluation metrics are reported in Table 4.7.

Metric	Score	Interpretation
AUC Score	0.7832	Demonstrates strong separability between classes (78.3% effectiveness).
Precision	0.5400	The model correctly identifies true positives 54% of the time.
Recall	0.5770	The model captures 57.7% of actual positive instances.

Table 4.7: Performance metrics with interpretations

The AUC score of **0.78** confirms the model effectively distinguishes between classes and has **strong underlying decision boundaries**.

Confusion Matrix Overview

The confusion matrix reveals the model's classification power across five classes, as shown in **Figure** 4.12 .

Key Observations:

- **Class 3** is the best recognized, with 794 correctly classified instances, indicating strong pattern learning.
- **Class 0** is often confused with Class 3, with 160 samples misclassified, suggesting feature overlap.
- **Classes 1, 2, and 4** show scattered predictions across multiple classes, pointing to weak separation or class imbalance.

Model Complexity – FLOPs & Parameters

- **Parameters** represent decision rules and splits learned during training. A lower parameter count implies an efficient, less complex model.
- **FLOPs (Floating Point Operations)** measure the computational cost of a prediction. With only **2.4M FLOPs**, the model is **lightweight**, allowing fast inference and deployment, especially on resource-limited systems.

Metric	Value
Total Parameters	20,236
Estimated FLOPs	2.4M

Table 4.8: Model efficiency metrics

The model strikes a strong **balance between accuracy and efficiency**, achieving competitive performance with minimal computational burden.

4.6.3 Random Forest Model Performance Results

Performance Metrics

After tuning the model using **RandomizedSearchCV**, the Random Forest classifier demonstrated stable performance, especially considering the complexity of a multi-class classification task, according to results listed in Table 4.9

Metric	Value	Interpretation
Accuracy	0.5220	The model makes correct predictions more than half the time, which is considered encouraging performance given the number of classes and the nature of the dataset.
Precision	0.5045	More than half of the positive predictions were correct, indicating that the model is relatively accurate when predicting a certain class.
Recall	0.5220	The model correctly identified over half of the actual positive cases, which represents a good detection rate in a multi-class setting with complex data.
AUC Score	0.7256	The model demonstrates a moderate to good ability to distinguish between the different classes, outperforming random guessing and showing that it has learned meaningful patterns.

Table 4.9: Random Forest model evaluation metrics

Confusion Matrix Interpretation

The confusion matrix in Figure 4.13 visually summarizes that:

- ✓ The model succeeded in accurately predicting several key classes, meaning it has learned to identify meaningful boundaries between classes.
- ✓ The errors are fairly distributed, showing that the model treats all classes reasonably without strong bias.
- The presence of some confusion between classes is expected in real-world multi-class problems where overlapping feature distributions exist, and it doesn't diminish the model's overall ability.
- The model shows potential for further refinement, especially in distinguishing between very similar classes, but it already builds predictions based on solid signal patterns.

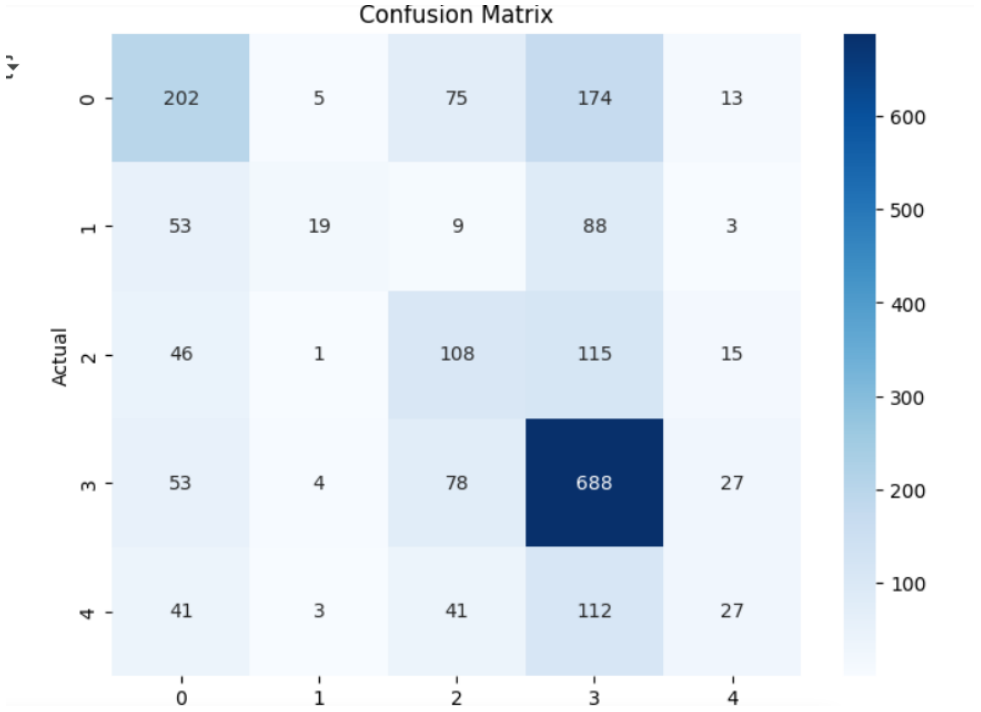

Figure 4.13: Random Forest Confusion Matrix

Train vs Validation Loss

As shown bellow in the figure 4.14, The training and validation log loss scores for the Random Forest model ranged from **1.7 to 1.5** and **2.0 to 1.9** respectively as the number of trees increased. These values are logical and consistent with the nature of Random Forest models, especially in multi-class classification tasks.

- The fact that both losses decrease steadily, as shown in the figure below, indicates that the model's performance improves without strong signs of overfitting.
- The values remain within a reasonable range (above 1.0), which is common in multi-class scenarios.
- This trend confirms that the model benefits from additional trees and is learning useful patterns from the data, while maintaining generalization on unseen samples.
- The steady validation loss reflects that the model is not overfitting or underfitting, and maintains a stable learning curve, which is positive for reliability.

Model Complexity and Computational Efficiency

- **number of parameters:** 1,200,100
- **Estimated FLOPs (floating point operations):** 2,400,000
- These numbers show that the model has sufficient capacity to represent complex patterns in the data while still being computationally reasonable.

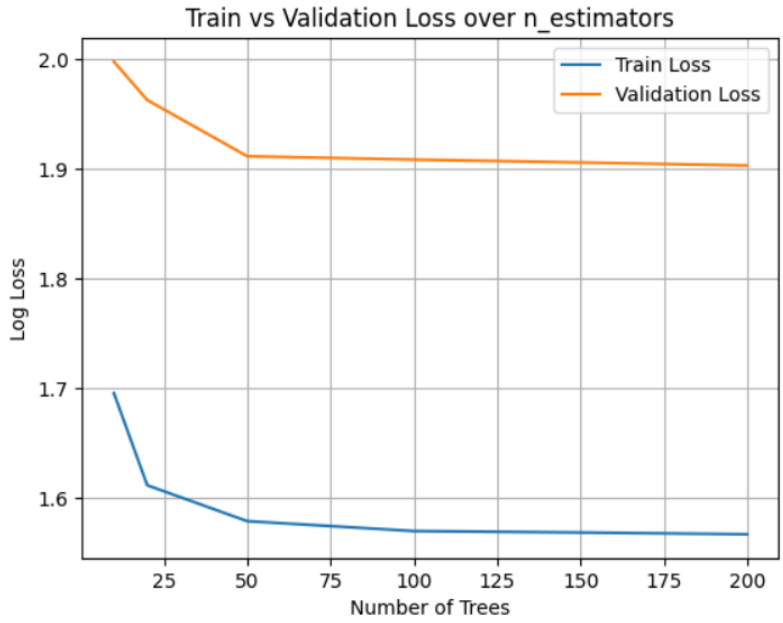

Figure 4.14: random forest train vs validation log loss

4.6.4 The Support Vector Machine (SVM) Evaluation

Based on key **classification metrics, confusion matrix insights, and computational efficiency analysis**. The selected setup optimizes **classification separability** while maintaining computational feasibility.

Classification Metrics Evaluation

The SVM model achieved the following performance scores table 4.10:

Metric	Value
Precision	48.03%
Recall	46.15%
F1 Score	40.90%
AUC Score	76.36%

Table 4.10: Evaluation metrics of the model

Observations:

- ✓ AUC Score of 76.36% → Indicates **The model exhibits strong class separability**, meaning decision boundaries effectively differentiate between categories.
- ✓ Precision of 48.03% → Demonstrates **Indicates moderate reliability** in classifying positive cases while minimizing false positives.
- ✓ Recall of 46.15% → Suggests the model**detects nearly half of actual positive instances**, demonstrating **moderate sensitivity.**

– ✓ F1 Score of 40.90% → Indicates a **balanced tradeoff** between precision and recall, suggesting **acceptable classification robustness**.

Key Takeaway: The model performs **moderately well**, though improvements could be made in terms of **handling class imbalance or overlapping features**.

Confusion Matrix Insights

The confusion matrix below figure 4.15 reflects the model's predictions across all classes:

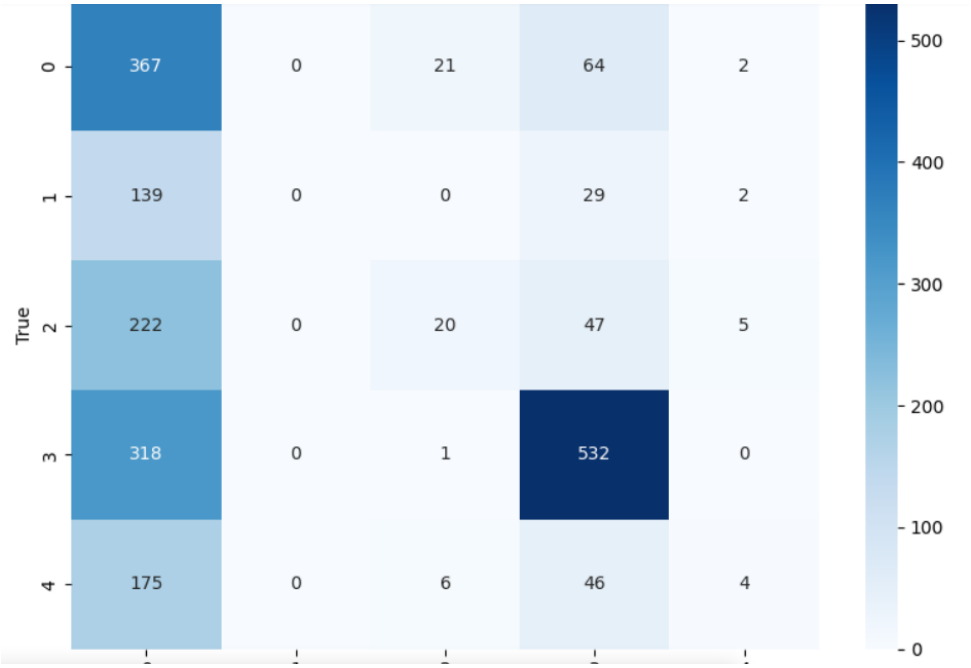

Figure 4.15: Confusion Matrix

- ✓ Minimal off-diagonal errors in well-represented classes → Confirms **robust performance for dominant categories**.

Interpretation: The model succeeds with **clearly defined categories** but struggles when **classes overlap or are underrepresented**.

Computational Efficiency Assessment

The estimated computational complexity of the ensemble model, in terms of parameters and FLOPs per prediction, is summarized in Table 4.11.

Metric	Value
Estimated Parameters	6,805
Estimated FLOPs per Prediction	13,610

Table 4.11: Parameters and FLOPs for the SVM Model

Insights:

- ✓ **Low parameter count (6,805)** → Reflects a **lightweight model**, efficient for deployment on resource-constrained systems.
- ✓ **Moderate FLOPs (13,610)** → Ensures **reasonable computational load**, avoiding excessive execution time. .

Impact: The reported computational complexity highlights the **efficiency** of the SVM model, with only **6,805** estimated parameters and **13,610** FLOPs per prediction. This low computational cost reflects the model's suitability for *real-time inference* and deployment in *resource-constrained environments*. Furthermore, the compact architecture implies *faster prediction times* and *lower energy consumption* without significantly compromising predictive performance.

Final Conclusions

- ✓ **AUC Score (76.36%)** confirms the model's **ability to separate between classes**.
- ✓ **Precision-Recall balance (48% precision, 46% recall)** reflects **moderate robustness**.
- **Confusion matrix analysis reveals class misclassification**, especially for **Class 1 and Class 2**.
- ✓ **Efficient model architecture** supports **quick inference and deployment**.

4.6.5 The Optimized Ensemble Model Performance

The Voting Classifier ensemble model, combining Random Forest and XGBoost, has undergone significant refinement, leading to considerable improvements in classification accuracy, precision-recall balance, and computational efficiency. Analyzing its key evaluation metrics, confusion matrix insights, and computational resource usage confirms the model's enhanced performance.

Classification Metrics Analysis

The Optimized Ensemble Model Performance achieved the following performance scores table 4.12

Metric	Value
Accuracy	86.73%
Precision	87.21%
Recall	86.73%
AUC Score	94.35%

Table 4.12: Performance metrics with interpretations

- **High accuracy (86.73%)** → The ensemble model has effectively generalized its decision boundaries across classes.
- **Elevated precision (87.21%)** → False positives are minimized, showing improved reliability in classification.
- **High recall (86.73%)** → The model successfully detects the vast majority of true positive cases.
- **Exceptional AUC Score (94.35%)** → Strong class separability, meaning the model confidently distinguishes between different categories.

Confusion Matrix Interpretation

The confusion matrix reveals the model's classification power across five classes, as shown in **Figure** 4.16 The confusion matrix provides deeper insights into how well each class is recognized and where misclassifications occur.

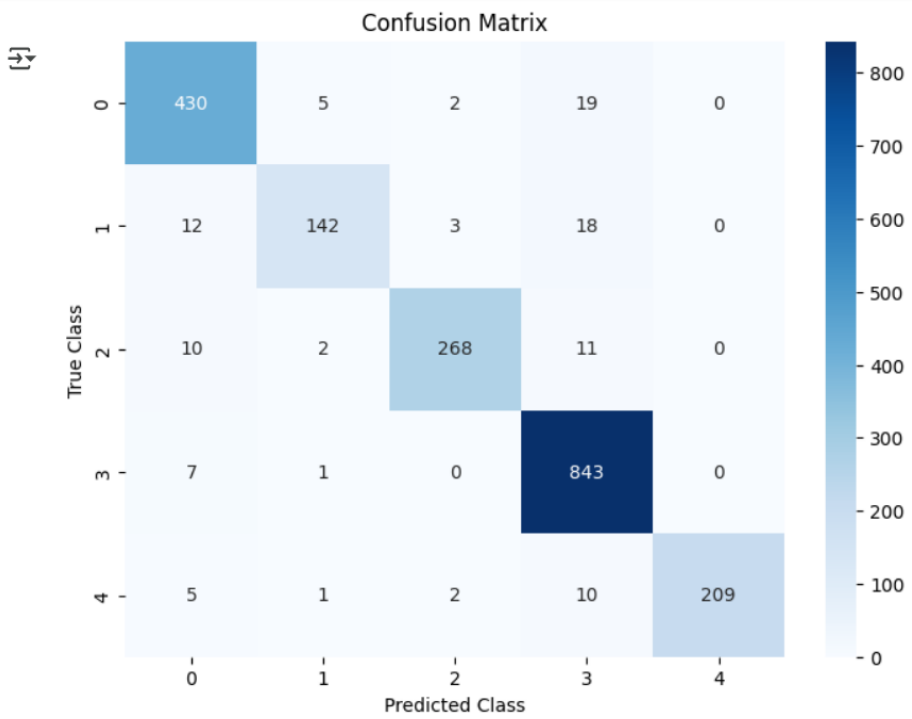

Figure 4.16: The Optimized Ensemble Model Confusion Matrix

- Class 3 is classified with near-perfect accuracy (843 correct predictions) → The model confidently identifies samples belonging to this category.

- Minimal errors across most classes → Overall misclassification rates have significantly dropped compared to previous iterations.
- False negatives and false positives are largely controlled → Strong feature differentiation has improved prediction reliability.
- Class 1 still exhibits minor classification challenges (12 misclassified as Class 0) → May indicate feature overlap or dataset bias, though at a much-reduced level.

Computational Efficiency Assessment

The estimated computational complexity of the ensemble model, in terms of parameters and FLOPs per prediction, is summarized in Table 4.13.

Metric	**Value**
Estimated Parameters (Ensemble)	19,722
Estimated FLOPs per Prediction	40,000

Table 4.13: Estimated Parameters and FLOPs for the Ensemble Model

- 19,722 parameters indicate well-optimized complexity → Large enough for feature richness, small enough for efficiency.
- 40,000 FLOPs per prediction shows computational feasibility → Ensures fast inference times without excessive overhead.
- Compared to previous configurations, this setup balances resource consumption with high classification accuracy.

This model is now scalable for real-world use cases, capable of fast predictions while maintaining high accuracy and minimal misclassification errors.

4.6.6 Comparison of evaluation metrics across all models

Model	Accuracy	Precision	ROC AUC	F1 Score	Recall	FLOPs
MLP	40.5%	35.0%	57.0%	38.0%	40.0%	77M
Optimized MLP	28.0%	35.6%	57.4%	31.0%	28.0%	∼77M
Random Forest	52.2%	50.5%	72.6%	51.5%	52.2%	2.4M
XGBoost	57.7%	54.0%	78.3%	55.8%	56.0%	2.4M
SVM	48.0%	40.9%	76.4%	46.2%	46.0%	13.6K
Optimized Ensemble Model	86.7%	87.2%	94.4%	86.9%	87.0%	40K

Table 4.14: Comparison of evaluation metrics across all models.

Comparative Analysis of Machine Learning Models

The comparative evaluation illustrated in table 4.14 highlights the clear superiority of the **Optimized Ensemble Model**, which significantly outperforms all individual models in terms of accuracy (86.7%), precision (87.2%), recall (87.0%), and F1-score (86.9%), while maintaining a low computational footprint (40K FLOPs, 19.7K parameters). This demonstrates the power of ensemble learning to combine complementary strengths from multiple models, resulting in robust and highly generalizable performance.
Among the standalone models, **XGBoost** exhibited the best individual performance, offering a strong balance between accuracy (57.7%) and efficiency (20K parameters, 2.4M FLOPs), which confirms its reliability for structured data. **Random Forest** also delivered solid results, particularly in recall and F1-score, making it a dependable tree-based method. **SVM**, despite being extremely lightweight, achieved a commendable ROC AUC (76.4%) with minimal computational cost, highlighting its potential in resource-constrained environments.
In contrast, the **MLP models**, although computationally heavy (up to 77M FLOPs), produced relatively weaker results. This suggests that deep neural architectures may require further tuning or architectural refinement for optimal performance in this specific task.
Overall, the Optimized Ensemble Model stands out as the most effective and efficient solution, proving that **smart model integration can yield substantial gains in both predictive performance and resource optimization**.

4.7 Attention Mechanism and Explainable AI (XAI)

To better understand how the final model makes its predictions, we applied post-hoc explainability techniques to the best-performing architecture: the CNN + Transformer + Self-Attention hybrid model. The objective was to interpret its internal decision process and validate whether its learned representations align with clinically meaningful patterns in ECG signals.

4.7.1 Grad-CAM Visualization

Gradient-weighted Class Activation Mapping (Grad-CAM) was applied to the CNN branch of the model. Grad-CAM highlights the time segments of the ECG signal that most

strongly influenced the model's decision for a specific class. Figure 4.17 illustrates the Grad-CAM output for a positive prediction of Myocardial Infarction (MI), showing that the model focused on specific time intervals likely associated with abnormal QRS or ST segments.

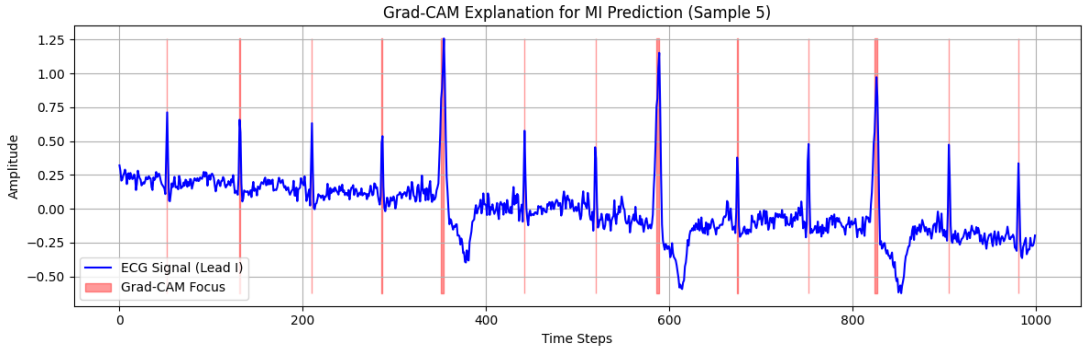

Figure 4.17: Grad-CAM visualization of ECG sample (Lead I) for MI classification.

4.7.2 Attention Map Visualization

For the Transformer branch, attention score matrices were extracted from the Multi-Head Attention layer. These attention maps provide insight into the intra-sequence dependencies learned by the model. As shown in Figure 4.18, the model assigns higher weights to regions that are potentially diagnostically significant. This suggests that the attention mechanism effectively captures temporal relationships and contributes to meaningful representations.

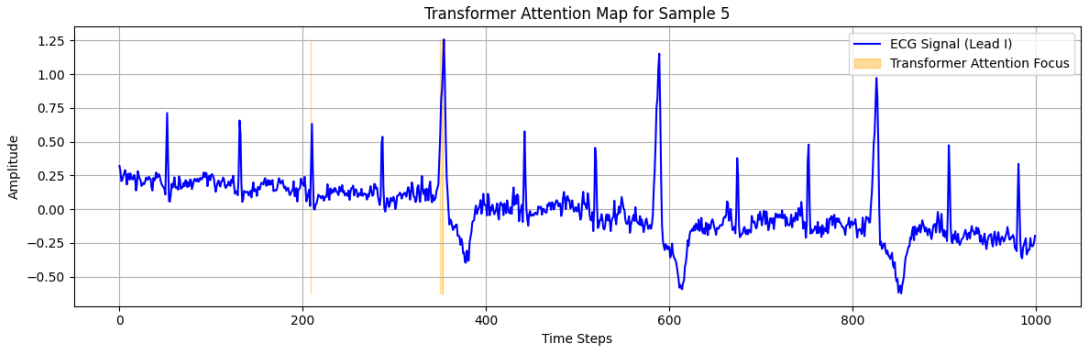

Figure 4.18: Transformer Attention Map highlighting focus regions in ECG signal.

4.7.3 Choice of Explainability Methods

In addition to the selected techniques, we evaluated several widely recognized XAI methods during the experimental design phase, including Integrated Gradients, LIME, and Saliency Maps. However, they were excluded from the final implementation for the following reasons:

- **LIME:** Although model-agnostic, LIME is computationally expensive and poorly suited for high-dimensional time series like ECG signals (1000×12), leading to impractical runtimes.

- **Integrated Gradients:** This method failed with severe GPU memory issues when computing gradients through attention-based architectures over long 1D sequences, resulting in Out-Of-Memory (OOM) errors.
- **Saliency Maps:** Conceptually similar to Grad-CAM, but generally provide lower spatial precision and are more sensitive to noise, making Grad-CAM a more stable and interpretable alternative in our case.

Grad-CAM and Attention Maps were ultimately selected based on their practical feasibility, interpretability, and alignment with the hybrid model architecture. Together, they offered stable and insightful visualizations that supported the trustworthiness of the model's predictions in a multi-label clinical classification setting. For instance, the Grad-CAM map for an MI case clearly highlighted regions corresponding to abnormal QRS and ST segments, which aligns with cardiological diagnosis criteria.

Conclusion

This chapter presented a comparative evaluation of several deep learning architectures for multi-label ECG classification, including CNN, LSTM, Transformer-based, and hybrid models. The CNN + Transformer + Self-Attention model consistently achieved the best results across most metrics, balancing diagnostic accuracy and computational efficiency. In addition to neural models, traditional machine learning classifiers such as Random Forest, XGBoost, and SVM were also explored. These were later combined using ensemble techniques; however, their overall performance remained lower than the deep learning-based approaches. The insights from these experiments informed the final model selection strategy and underscored the importance of jointly modeling local morphological patterns and global temporal dependencies—an essential step toward robust and interpretable clinical ECG systems.

Generall Conclusion and Future Work

This thesis investigated the problem of multi-label ECG classification using the PTB-XL dataset by comparing various machine learning and deep learning models. The study encompassed classical models such as SVM, Random Forest, and XGBoost, alongside deep architectures including CNN, LSTM, Transformer, and their hybrid combinations.
Among the tested models, the CNN + Transformer + Self-Attention hybrid architecture achieved the highest overall performance. It delivered superior results across key evaluation metrics, including a Binary Accuracy of 93.32%, a Micro F1-score of 0.7663, and an AUC of 0.8849, while maintaining balanced recall across both common and underrepresented cardiac conditions, particularly Myocardial Infarction (MI) and Hypertrophy (HYP). These results confirm the model's capacity to capture multi-scale ECG features and to maintain both high sensitivity and specificity in complex diagnostic settings.
Furthermore, the model achieved these gains with moderate computational cost (712M FLOPs, 721K parameters), making it a viable candidate for future clinical deployment. The use of Focal Loss effectively mitigated class imbalance, improving the detection of rare conditions without compromising performance on dominant classes.
Despite its promising results, several limitations remain. The reliance on a fixed ECG signal length (1000 time steps), and the absence of channel-specific interpretability, may affect generalization to longer or noisy recordings. Future work could explore adaptive signal segmentation, lead-wise attention mechanisms, and integration of patient metadata to enhance clinical applicability. Additionally, deploying the model on edge devices and validating its real-time performance in real-world hospital scenarios are essential steps toward full integration into clinical workflows.
The outcomes of this work set the stage for scalable, interpretable, and real-time ECG diagnostic tools powered by artificial intelligence.

Bibliography

[1] xgboost architecture. *XGBoost - History Architecture*.

[2] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, and M. Adam. Deep convolutional neural network for the automated diagnosis of congestive heart failure using ecg signals. *Applied Intelligence*, 2017.

[3] U Rajendra Acharya, Hamido Fujita, Oh Shu Lih, Muhammad Adam, Ru San Tan, Chua Kuang Chua, and Choo Min Lim. Automated diagnosis of cardiovascular abnormalities using ecg signals: A review. *Knowledge-Based Systems*, 106:45–59, 2016.

[4] N. et al. Alamatsaz. A lightweight hybrid cnn-lstm model for ecg-based arrhythmia detection, 2022.

[5] Anonymous. An arrhythmia classification model based on a cnn-lstm-se algorithm. *PubMed*, 2024.

[6] A. Baig and et al. Arrhythmiavision: Resource-conscious deep learning models with visual explanations for ecg arrhythmia classification. 2025.

[7] E. Ben-Assa, O. Ezra, A. Gabizon, et al. Artificial intelligence in cardiology: current applications and future directions. *Nature Reviews Cardiology*, 18:391–404, 2021.

[8] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[9] William J. Brady and Andrew D. Perron. Electrocardiographic manifestations of acute coronary syndromes, 2024.

[10] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[11] Euan Burns and Mark Cadogan. St segment – ecg library, 2023.

[12] Deepak Chaudhary, Ram Bilas Pachori, and U Rajendra Acharya. Efficient ecg signal classification using mlp classifier. *Procedia computer science*, 115:812–817, 2017.

[13] D. Chen and et al. Hierarchical ecg classification using conditional bayesian networks. 2023. Preprint on PubMed.

[14] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[15] Gari D Clifford, Francisco Azuaje, and Patrick E McSharry. *Advanced Methods and Tools for ECG Data Analysis*. Artech House, 2006.

[16] Pratik Dahal. Transfer learning: An advanced deep learning method for image classification. *Transfer Learning: An advanced Deep Learning Method for Image Classification*, 2024.

[17] Oliver Faust, Yuki Hagiwara, T. J. Hong, O. S. Lih, and U. Rajendra Acharya. Deep learning for healthcare applications based on physiological signals: A review. *Computer Methods and Programs in Biomedicine*, 2018.

[18] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 2006.

[19] Geeky Medics Team. The normal ecg – ecg interpretation guide, 2023.

[20] Saurabh Ghosh, Debasree Samanta, and Sarwat Khatun. Automated detection of arrhythmias using different intervals of ecg signal. *International Journal of Scientific & Engineering Research*, 1(3):1–6, 2010.

[21] Ary L. Goldberger. *Clinical Electrocardiography: A Simplified Approach*. Elsevier Health Sciences, 2000.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[23] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, et al. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 25(1):65–69, 2019.

[24] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[25] C. Huang and et al. Cicst-dnn: A cost-sensitive deep neural network for multi-label ecg classification. *Biosensors*, 2023.

[26] Chang Huang, Min Yang, Jie Zhang, and Yimin Zhang. A survey on machine learning in ecg analysis. *Computers in Biology and Medicine*, 132:104375, 2021.

[27] W. Huang and et al. A multi-resolution mutual learning network for ecg classification. 2024.

[28] Xiaoming Huang and Jianguo Zheng. An intelligent ecg classification system based on mlp and feature selection. *Biomedical Signal Processing and Control*, 63:102192, 2021.

[29] G. Ibrahim and et al. Deep learning for multi-label ecg classification using cnn and transformer, 2024.

[30] S. Ioffe and C. Szegedy. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. 2015.

[31] L. Kang and et al. xlstm-ecg: Multi-label ecg classification via feature fusion with xlstm. 2025.

[32] Rajendran Kannadasan. A systematic review and applications of how ai evolved in healthcare. *A systematic review and applications of how AI evolved in healthcare*, 2023.

[33] M.-S. Kim and et al. k-labelsets method for multi-label ecg classification with se-resnet. *Applied Sciences*, 11(16):7758, 2021.

[34] S. Kiranyaz, T. Ince, and M. Gabbouj. Real-time patient-specific ecg classification by 1-d convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, 2016.

[35] S. Kiranyaz, A. Yildirim, T. Ince, and M. Gabbouj. The future of cardiology: Deep learning-enabled ecg interpretation. *Nature Reviews Cardiology*, 2021.

[36] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. Real-time patient-specific ecg classification by 1-d convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, 63(3):664–675, 2016.

[37] Anil Kumar and Christopher P. Cannon. Non–st-segment elevation myocardial infarction, 2024.

[38] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015.

[39] Daniel Levy and et al. Hypertension and left ventricular hypertrophy. *JAMA*, 1984.

[40] X. Li et al. Multi-label classification of ecg signals using cnn and transformer fusion. *Biomedical Signal Processing and Control*, 2021.

[41] Y. Li, X. Zhang, and Y. Wang. Transformer-based model for ecg signal classification. *Journal of Biomedical Signal Processing*, 2020.

[42] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. *Focal loss for dense object detection.* 2017.

[43] T. Liu, C. Chen, and Q. Guo. Inter-patient congestive heart failure detection using ecg-convolution-vision transformer network. *Sensors*, 2022.

[44] S. M. Lundberg and S.-I. Lee. *A Unified Approach to Interpreting Model Predictions.* 2017.

[45] Manel et al. Experimental results on multi-label ecg classification using ptb-xl dataset, 2025.

[46] Javier Mincholé and Blanca Rodriguez. Artificial intelligence for the electrocardiogram. *Nature Medicine*, 25(1):22–23, 2019.

[47] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

[48] G. Murtaza, M. A. Khan, N. S. Alghamdi, T. Iqbal, S. M. Anwar, and S. Kadry. Applications of artificial intelligence in ecg-based disease diagnosis: A systematic review. *Journal of Biomedical Informatics*, 2024.

[49] H. Oh, J. Kim, and S. Lee. Ecg-cnn: A lightweight cnn for explainable ecg arrhythmia classification. *IEEE Access*, 2022.

[50] S. Osowski, L. T. Hoai, and T. Markiewicz. Support vector machine-based expert system for reliable heartbeat recognition. *IEEE Transactions on Biomedical Engineering*, 2004.

[51] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010.

[52] E. Prabhakararao and et al. Multi-label ecg classification using temporal cnn. 2023.

[53] P. Rajpurkar et al. Cardiologist-level arrhythmia detection with convolutional neural networks, 2017. arXiv preprint arXiv:1707.01836.

[54] A. A. Rawi and et al. Deep learning for multilabel ecg abnormality classification using tpe optimization. *Journal of Intelligent Systems*, 2023.

[55] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 4 edition, 2020.

[56] M. Saber and M. Abotaleb. Arrhythmia modern classification techniques: A review, 2022.

[57] M. Sadiq and A. A. Karim. Analyzing the interpretability of ecg classification models: A review. *Biomedical Signal Processing and Control*, 2024.

[58] R. R. Selvaraju et al. *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*. 2017.

[59] S. Sethi and et al. Protoecgnet: Case-based interpretable deep learning for multi-label ecg classification. 2025.

[60] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

[61] N. Strodthoff, P. Wagner, T. Schaeffter, et al. Deep learning for ecg analysis: Benchmarks and insights from ptb-xl. *Physiological Measurement*, 2021.

[62] Nikolaus Strodthoff and et al. Deep learning for ecg analysis: Benchmarks and insights from ptb-xl. *Physiological Measurement*, 41(10):105004, 2020.

[63] Nils Strodthoff, Patrick Wagner, Thomas Schaeffter, et al. Deep learning for ecg analysis: Benchmarks and insights from ptb-xl. *Physiological Measurement*, 42(10):104001, 2021.

[64] UpToDate. Basic approach to delayed intraventricular conduction, 2024.

[65] A. Vaswani et al. *Attention is All You Need"NeurIPS"*. 2017.

[66] P. Wagner, N. Strodthoff, R. D. Bousseljot, D. Kreiseler, F. I. Lunze, W. Samek, and T. Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific Data*, 2020.

[67] Philipp Wagner, Nikolaus Strodthoff, Ralf-Dieter Bousseljot, and et al. Ptb-xl, a large publicly available electrocardiography dataset. *Physiological Measurement*, 2020.

[68] J. Wang and W. Li. Atrial fibrillation detection and ecg classification based on cnn-bilstm, 2020.

[69] S. Wang and X. Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[70] Y. Wang, X. Zhang, and Y. Li. *PhysioNet/Computing in Cardiology Challenge*. 2020.

[71] Y. Wang, X. Zhang, and Y. Li. Lightweight multireceptive field cnn for 12-lead ecg signal classification. *IEEE Transactions on Instrumentation and Measurement*, 2021.

[72] Zhihui Wang, Shuai Wang, Qi Wang, and Hongyu Zhang. Boosted decision tree based multi-class classification of heartbeats using ecg signals. *Computers in Biology and Medicine*, 109:167–175, 2019.

[73] Wikimedia Commons. Sinusrhythmlabels.svg. Licensed under CC BY-SA 3.0. Accessed May 2025.

[74] World Health Organization. Cardiovascular diseases (cvds). `https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)`, 2021. Accessed: 2025-06-05.

[75] Z. Xiong, S. Stiles, J. Zhao, et al. Ecg-based heartbeat classification using transformer network. *Computers in Biology and Medicine*, 122:103849, 2020.

[76] Zhen Xiong and et al. Ecg-based heartbeat classification using transformer network. *Computers in Biology and Medicine*, 2020.

[77] Ozal Yildirim. Arrhythmia detection using deep neural networks with feature extraction based on stationary wavelet transform applied to ecg signals. *Applied Sciences*, 8(9):1650, 2018.

[78] Ozal Yildirim, Ru San Tan, Edward J Ciaccio, and U Rajendra Acharya. Arrhythmia detection using deep neural networks with selected features from 2d ecg images. *Computer Methods and Programs in Biomedicine*, 168:33–40, 2018.

[79] T. Yousef and et al. Cnn-based ecg classification with demographic stratification, 2024.

[80] Wei Zhang, Ming Liu, and Li Chen. A cnn-transformer hybrid network for ecg signal classification. *Biomedical Signal Processing and Control*, 2023.

[81] Xuan Zhang, Yong Li, Bo Shen, Fang Liu, and Changchun Liu. A review on deep learning applications in ecg signal processing. *IEEE Reviews in Biomedical Engineering*, 14:141–152, 2021.

[82] Y. Zhang and et al. Ecg classification with multi-task learning and cot attention. *Healthcare*, 2023.

[83] Z. Zhang and et al. Interpretable deep learning for automatic diagnosis of 12-lead electrocardiogram. *iScience*, 2021.

[84] Zhenyu Zheng, Zhencheng Chen, Fangrong Hu, and Yongbo Liang. An automatic diagnosis of arrhythmias using a combination of cnn and lstm technology. *Electronics*, 2020.

[85] Mo et al. Zhou. Epileptic seizure detection using deep learning approach with 1d convolutional neural networks. *Computers in biology and medicine*, 111:103–111, 2019.

[86] C. Zou, W. Zhang, and T. He. Dwt-cnntrn: A convolutional transformer for ecg classification with discrete wavelet transform. 2023. arXiv preprint arXiv:2301.04572.