Enhancing Cross-Hospital Generalizability of Deep Learning Models in ECG Classification: A Comparative Study

University of Turku
Department of Computing
Master of Science (Tech) Thesis
Data Analytics
June 2025
Aituar Bektanov

Supervisors:
Antti Airola
Tapio Pahikkala

UNIVERSITY OF TURKU
Department of Computing

Aituar Bektanov: Enhancing Cross-Hospital Generalizability of Deep Learning Models in ECG Classification: A Comparative Study

Master of Science (Tech) Thesis, 84 p.
Data Analytics
June 2025

Deep learning models have demonstrated excellent performance in electrocardiogram (ECG) classification tasks. However, because of domain shift caused by differences in patient demographics, recording devices, and labeling procedures, its generalizability across data from other hospitals is still limited. Therefore, this thesis examines whether some domain generalization methods are useful for improving the cross-hospital generalizability of ECG classification models.

The thesis investigates two domain generalization methods: multi-source domain-adversarial training (DANN) and MixStyle. Both of them are implemented inside an SE-ResNet model architecture. The models are trained, validated, and tested on a multi-source dataset comprising five publicly available ECG data sources. The macro-averaged area under the ROC curve (AUROC) serves as an evaluation metric of the performance of the models on unseen domains.

According to the results of the experiments, the DANN-based model modestly outperforms the baseline model in several cases, particularly when the CPSC and CPSC-Extra and the SPH domains are used as test sets. On the other hand, the MixStyle-based model does not produce improved generalization results. Additionally, the results suggest that standard hyperparameter selection using the validation set may not work well for domain generalization in this context because validation sets do not contain data from unseen domains.

Overall, the results of this thesis show how complex the domain generalization problem in ECG classification is. Apart from that, they demonstrate that, even though multi-source domain-adversarial training (DANN) might be useful for improving the generalization performance of deep learning models in the context of ECG classification, it is not a standalone solution. Thorough future work, including the usage of other domain generalization methods and data from new domains, should be done on this topic.

Keywords: ECG, classification, multi-source, deep learning, domain generalization, cross-hospital, domain-adversarial neural networks (DANN), MixStyle

Contents

List of Figures

List of Tables

1 Introduction

An electrocardiogram, or ECG (sometimes also called an EKG), is a simple non-invasive test, which is used for recording the electrical activity of the heart [1]. In other words, an ECG is essentially the recording of any electrical changes that take place inside the heart during a cardiac cycle [2]. It is done by placing electrodes on the body. The ECG was invented in 1902 by a Dutch physician named William Einthoven [3]. Knowing how the ECG works and what it means helps to understand the functioning of the heart. Figure 1.1 illustrates a typical normal ECG.

The ECG is produced by the electrocardiograph, which refers to a machine that detects and picks up the electrical signals generated by the heart muscle during a contraction and relaxation cycle. The main principle of how the electrocardiograph operates is that a muscle, which contracts, produces a small electric current that can be detected and quantified via electrodes suitably positioned on the body. These electrodes are attached to the patient's skin through the use of a special jelly. When

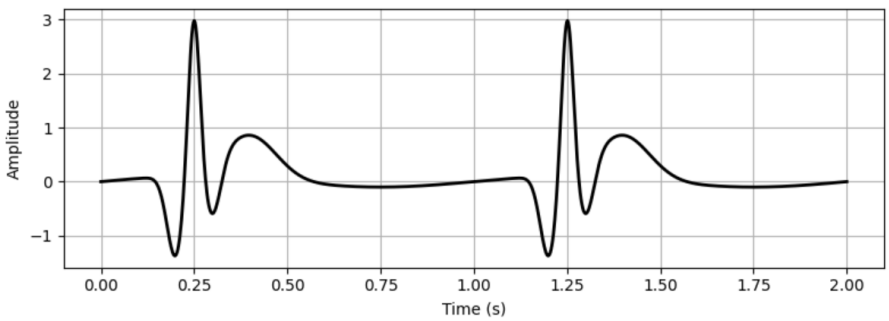

Figure 1.1: Illustration of a normal ECG. Adapted from Online Biology Notes [2]

the patient is in a supine and resting position in the medical chair, the electrodes are positioned on the legs, arms, and six different places on the chest above the heart region. Each electrode picks up the electric current and sends it to an amplifier inside the electrocardiograph, which subsequently amplifies the current and captures it as a wavy line on paper. In the electrocardiograph, there is a sensitive lever that tracks changes in current on a moving piece of paper. Sometimes, there can also be an oscilloscope, connected to a contemporary electrocardiograph, that is used to display the electric current on a screen [2].

An ECG lead is a visual description of the heart's electrical activity that is produced by reading several electrodes. Hence, each ECG lead is obtained via the analysis of the electrical currents recorded by several electrodes. There are a lot of ECG lead systems that have been tested and used, but the 12-lead ECG is the standard configuration that remains the most widely used and the most significant one. Indeed, the 12-lead ECG system has been shown to provide excellent possibilities for cardiac abnormality and condition diagnosis, as well as a thorough assessment of the heart's function. The 12-lead ECG displays 12 leads that are generated through the use of 10 electrodes. It consists of three regular limb leads (I, II, and III), three augmented limb leads (aVR, aVL, aVF), and six chest (precordial) leads (V1-V6) [1].

The ECG has been a very important tool in healthcare. It is regarded as one of the most widely used medical tools, with around 200 million recordings captured annually worldwide [4]. There are plenty of different purposes and applications for which the ECG is used. One of the earliest clinical applications of the ECG was its ability to detect acute myocardial injury [5]. This application made it easier for clinicians to distinguish between non-cardiac chest pain mimickers and actual cardiac chest pain. Currently, one of the most common clinical ECG applications is arrhythmia detection [6]. Apart from that, there are other clinical utilities of the

ECG that include, but are not limited to, the detection of tachycardia, bradycardia, heart failure, congenital heart disease, and rheumatic heart disease [3]. There are many other cardiac abnormalities and cardiovascular diseases (CVD) affecting heart function that the ECG is able to detect and help prevent. The ECG is also used for the assessment of heart valve integrity, coronary blood flow, and metabolic disorders, as well as monitoring of some cardiac medications [3]. Moreover, the ECG can be found in the field of sports, where it is used in a physical exam to detect or rule out the presence of cardiomyopathy [7]. These examples are just a few of the many ECG applications. Overall, the ECG's broad use and powerful diagnostic capabilities in healthcare are made possible by its quick, non-invasive, and affordable nature [6].

1.1 Usage of Artificial Intelligence in ECG Analysis

Recently, Artificial Intelligence (AI) has been widely used in ECG analysis. AI models are being developed and utilized for many ECG analysis tasks, including the identification of ECG diagnoses and the forecast of different health disorders [8]. In particular, Deep Learning (DL), which is a branch of Machine Learning (ML) that uses multi-layered neural networks to imitate the learning process of the human brain, has been effectively utilized for the detection and classification of arrhythmia and other cardiac abnormalities, proving to perform better than average cardiologists [9] and even traditional ML methods [10]. These DL models decrease the need for manual ECG assessment and facilitate early detection of cardiac diseases. Recent studies and experiments have shown that DL models not only can interpret ECG automatically, but also extract and analyze raw data, making it possible to provide information that is not visible to the human eye and therefore beyond classic interpretation [11].

Deep Learning (DL) models are able to extract and analyze the information in ECG time series, and this can be used for enhancing the detection and classifi-

cation of cardiac diseases. These abilities have substantially advanced because of rapid developments in DL models and higher computational power and speed [10]. Some DL methods remove the necessity for manual feature selection and extraction, thus enabling automated feature selection and improved performance [12]. Moreover, DL models are able to comprehend and differentiate the different types of arrhythmia due to their intrinsic capabilities of detecting and interpreting temporal fluctuations in ECG signals [13]. Additionally, some of the DL algorithms, including Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and Transformers, are capable of comprehending both long-term anomalies across multiple heartbeats and short-term patterns inside individual heartbeats [14]. This property makes it possible for DL models to identify disorders like atrial fibrillation and premature ventricular contraction, which rely on single heartbeats and may need pattern recognition spanning multiple heartbeats [15].

1.2 Challenges in Cross-Hospital Generalizability of Deep Learning Models in ECG Classification

Despite the many benefits that AI models have to offer for healthcare, there are still several challenges in terms of widespread adoption and incorporation into medical procedures. Some of these challenges include data bias, restricted data access, and regulatory issues [16]. Despite these challenges, considerable advancements have been achieved in developing automated models for ECG classification [10]. However, there are still challenges related to model generalizability that AI, and in particular, deep learning models, face nowadays [17].

Model generalizability concerns exist mainly due to the fact that, in practice, test data is sometimes collected from another source, e.g., a different hospital, so both the training and test data come from different distributions [18]. In this case,

ECG classifiers trained on single-source data do not work because they produce overly optimistic results and raise unreasonable expectations [19]. One possible way to address such problems is to adopt multi-source data, which corresponds to the data that is gathered from multiple sources, such as several hospitals. The ability of a model to generalize may be reduced if it is trained on data from a single source, which may limit the model's exposure to the variability present in broader patient populations. Therefore, it is important to train models on diverse data sources to improve model generalizability across various domains, e.g., hospitals [20].

There are many studies on this that have been carried out in recent years. However, they all showed that these diagnostic deep learning models trained on multi-source hospital data still suffer from a severe performance drop when deployed to data from new data sources that were not used in the training and validation phases. In other words, these models fail to generalize to new, unseen hospitals, and using multi-source data for training is not enough for improving cross-hospital generalizability.

There are many reasons why deep learning models encounter challenges in achieving cross-hospital generalizability. First, each hospital corresponds to a distinct data source, and there is a distribution discrepancy between these sources. This discrepancy happens because different healthcare settings employ diverse recording protocols and devices with varying characteristics, including different sampling frequencies and signal gains [21]. Furthermore, different hospitals vary in terms of population differences, such as different demographics and health conditions [22]. Apart from that, different healthcare settings sometimes vary in terms of unobserved confounders and deployment environments [18]. Finally, hospitals may differ from each other in terms of distribution drifts over time, which means that there could be modifications to clinical procedures and changing diagnostic guidelines [23].

1.3 Importance of Addressing Domain Shift Issues in ECG Classification

Addressing these domain shift issues in clinical use, including ECG classification, is important to ensure that deep learning models provide reliable, accurate, and clinically applicable results across diverse hospitals and healthcare settings. Nowadays, deep learning models used in clinical settings should be generalizable to different hospitals because, most of the time, data from many hospitals is not available for training, so there is a need for models trained on data from other data sources. The generalizability of deep learning models has the potential to make them more robust, and thus, more accepted and effective in real-world clinical scenarios.

As discussed earlier, to improve the generalizability of diagnostic deep learning models in clinical settings, it is important to incorporate data from multiple hospitals into training. However, as discussed in Section 1.2, using data from multiple sources is not enough. There is still a drop in model performance when they are tested on data from unseen, new hospitals. A number of studies have suggested that this issue can, to an extent, be mitigated using the so-called multi-source domain-adversarial training (DANN). Hence, this thesis explores this technique in detail and implements and tests it on an ECG-classification task.

1.4 Thesis Aim and Research Questions

The **aim of this thesis** is to improve the cross-hospital generalizability of deep learning models used for electrocardiogram (ECG) classification. The study draws on previous research and model architectures, particularly the SE-ResNet model originally proposed by Zhao et al. [24] and later refactored and applied by Leinonen et al. [16], and makes use of publicly available benchmark ECG datasets. Within the current model architecture, two different domain generalization methods are ex-

plored and implemented: multi-source domain-adversarial training (DANN), which encourages the model to learn domain-invariant feature representations by adversarially aligning feature distributions across multiple source domains [25], and MixStyle, a data augmentation method that improves robustness by randomly mixing feature statistics (means and variances) across training samples to simulate unseen domains during training [26].

The performance of the baseline and the adapted models is then evaluated and compared to determine how well these generalization strategies aid the models' performance when applied to data from unseen domains (hospitals).

The study mainly focuses on multi-source domain-adversarial training (DANN) for improving the model's generalizability, since this method has been widely advertised by many studies. The MixStyle method is of secondary interest, and it is implemented and tested to provide some comparison to multi-source domain-adversarial training (DANN).

Overall, the thesis is guided by the following **research questions**:

1. Are domain generalization methods from recent literature, such as multi-source domain-adversarial training (DANN) and MixStyle, promising for improving generalizability in ECG classification?

2. To what extent do these domain generalization techniques improve the performance of deep learning models when evaluated on data from previously unseen sources?

3. How do factors such as the amount of training data and hyperparameter settings influence the effectiveness of these generalization methods?

1.5 Thesis Contents

The remaining chapters of the thesis are organized as follows:

Chapter 2 reviews relevant literature on deep learning for ECG classification and domain generalization methods, such as multi-source domain-adversarial training (DANN) and MixStyle. **Chapter 3** presents the materials and methods used in this study, including the datasets, model architectures, model evaluation measures, and experimental setup. **Chapter 4** presents the results of the experiments and compares the performance of the baseline and adapted models. **Chapter 5** provides a discussion of the findings, their implications, limitations, comparison to previous work, critical analysis, and future directions. Finally, **Chapter 6** concludes the thesis by summarizing the main findings.

2 Related Work

This chapter explores some previous work that is related to the thesis aim. In the first subchapter, there is a detailed overview of existing deep learning models for ECG classification, what results they produced, their performances on single-source and multi-source data, their challenges with testing on data from new hospitals, etc. The second subchapter contains a review of works that used a few identified promising approaches for domain generalization, their results, and the comparison of their performance with that of the baseline model.

The second subchapter intends to answer the **first research question**, which is: "Are domain generalization methods from recent literature, such as multi-source domain-adversarial training (DANN) and MixStyle, promising for improving generalizability in ECG classification?"

2.1 Overview of Existing Deep Learning Models for ECG Classification

This subchapter presents related work on the use of deep learning models for ECG classification when tested on both single-source and multi-source data. In order to find relevant studies, a literature search was carried out in May 2024 using Google Scholar, IEEE Xplore, PubMed, and Web of Science. The following search strings were used: "deep learning" AND "ECG classification", "ECG classification" AND

"deep learning" AND "multi-source", "ECG classification" AND "deep learning" AND "single-source", "multi-source training" AND "single-source training" AND "ECG", and "deep learning" AND "ECG classification" AND "domain shift". The review was limited to peer-reviewed journal and conference papers published between 2017 and 2024. The selection of studies was based on their relevance to ECG classification across single and multiple data sources.

2.1.1 Models Trained on Single-Source Data

High Performance of Deep Learning Models on Single-Source Data

There have been many studies on the use of deep learning models in ECG classification. Many well-known deep learning architectures have been successfully used in these studies. For example, in 2019, Hannun et al. [27] developed a deep neural network (DNN) to classify 12 ECG rhythm classes using a novel ECG dataset that they constructed themselves. This dataset contained 91,232 ECG records from 53,549 patients. In the end, the model attained an average area under the receiver operating characteristic curve (AUROC) of 0.97. What is more, the model achieved an average F1 score, corresponding to the harmonic mean of precision (positive predictive value) and recall (sensitivity) [28], of 0.837, which is greater than the score of average cardiologists (0.780).

Similarly, in 2017, Zhang et al. [29] proposed and used a patient-specific electrocardiogram (ECG) classification algorithm based on recurrent neural networks (RNNs). The algorithm resulted in an average accuracy of 99.40%. The study used data from the MIT-BIH Arrhythmia database, which comprises ECG recordings collected from a population of patients at Boston's Beth Israel Hospital [30]. The same data was used by Ö. Yildirim [31] in 2018 when he implemented an algorithm based on bidirectional long-short-term memory networks (LSTMs), resulting in an accuracy of 99.39%. In addition, Xia et al. [32], in 2017, proposed and explored deep

convolutional neural networks (DCNNs) for automatic detection of atrial fibrillation, achieving an accuracy of 98.63%.

There are many other similar studies using different deep learning models that have shown superior performance. Ansari et al. [10] made an overview of the progress of deep learning for ECG arrhythmia detection and classification for the period 2017-2023. They analyzed many different deep learning models and concluded that many of them perform great in ECG classification. Table A.1 from Appendix A shows an overview of different deep learning models for the detection and classification of ECG arrhythmias by Ansari et al. [10]. Most of these models demonstrate high values for accuracy, sensitivity, and specificity.

Limitations of Single-Source Training for Generalization

However, these models have been trained on data from only one hospital, so they are not useful for improving cross-hospital generalizability of deep learning models in ECG classification. When a deep learning model is trained on data from only one hospital, its performance substantially drops when it is tested on data from another hospital.

There are several studies that empirically demonstrate a big performance drop when, in the context of ECG classification, a deep learning model trained on data from one source is tested on data from a new, unseen source. For example, in 2023, Avetisyan et al. [33] trained a DenseNet model from the family of CNNs on two datasets - TIS (private) and PTB-XL (public). These datasets represent two different data sources. When the model was trained and tested on the TIS dataset, the F2-score was 0.917 when predicting the presence of atrial fibrillation. However, when the model was trained on the PTB-Xl dataset and tested on the TIS dataset, the F2-score dropped to 0.832. Similarly, when predicting the presence of left bundle branch block, the model trained and tested on the PTB-XL dataset yielded an F2-

score value of 0.762, while the model trained on the TIS dataset and tested on the PTB-XL dataset yielded an F2-score value of 0.599. The F2 score is defined as the harmonic mean of precision and recall, where recall is given twice as much weight as precision [34].

As discussed before, to address the problem of performance drop when testing on data from a new hospital or source, there have been some studies that used multi-source data for training ECG classification models. For example, Leinonen et al. [16] conducted an empirical study to investigate multi-source cross-validation methods in clinical ECG classification.

2.1.2 Models Trained on Multi-Source Data

Improved Evaluation with Leave-Source-Out Cross-Validation

In their study, Leinonen et al. [16] fetched ECG data from 5 different data sources and used 4 of them for training several deep learning and machine learning models, and then testing these models on the fifth data source. Also, in their study, apart from using multi-source data, Leinonen et al. [16] used the so-called leave-source-out cross-validation method when evaluating the accuracy of deep learning and machine learning models used for ECG classification tasks. This was done to obtain a more comprehensive and realistic evaluation of DL and ML-based ECG classifiers on a new, unseen data source and of the overall generalizability of the model. In the leave-source-out cross-validation method, in contrast to the standard K-fold cross-validation method, each fold contains data from a distinct data source or domain [16].

As a result, Leinonen et al. [16] demonstrated that leave-source-out cross-validation produces performance estimates that are more reliable and almost bias-free, while "K-fold cross-validation systemically overestimates prediction performance when the end goal is to generalize to new sources" [16].

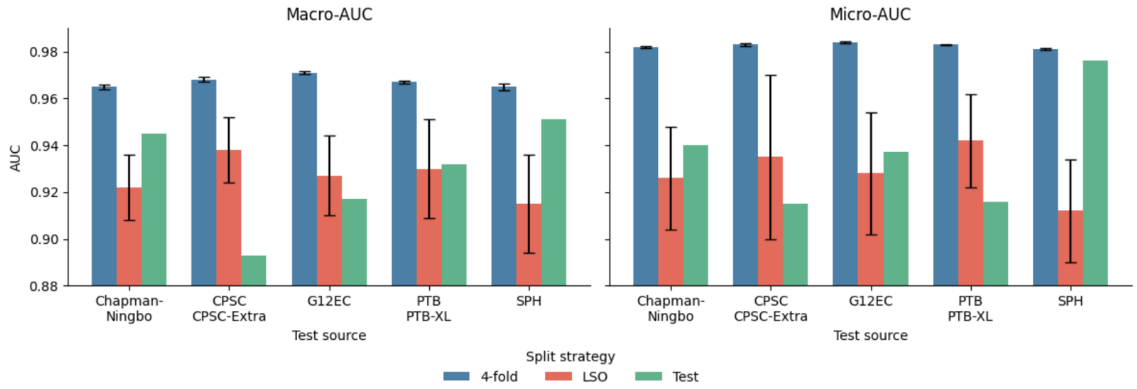

Figure 2.1: AUROC (± SD) scores for the 4-fold cross-validation, the leave-source-out cross-validation, and the test results per test source by the ResNet model, from the study by Leinonen et al. [16]. Adapted from Leinonen et al. [16]

Leinonen et al. [16] used two deep learning models, namely a residual neural network (ResNet) architecture with additional squeeze and excitation (SE) blocks and a wide and deep transformer neural network, and two traditional machine learning models, namely a logistic regression (LR) model and a XGBoost model. The ResNet model resulted in a better performance on average compared to the other models across all data sources as test sets. Figure 2.1 shows the resulting macro-averaged and micro-averaged areas under the ROC curve (AUROC) for the 4-fold cross-validation, the leave-source-out cross-validation, and the test results per test source by the ResNet model. The areas under the ROC curve (AUROC), including both the micro-averaged and macro-averaged ones, are explained in detail in Section 3.5.

ResNet Performance and Generalization Gap

From the figure, it can be seen that, with the exception of the SPH dataset, the leave-source-out cross-validation estimates seem to better match the test results than 4-fold CV estimates from any other test source, meaning that the results from the leave-source-out cross-validation are more reliable than those from the K-fold

cross-validation when it comes to the cross-hospital generalizability of the model.

The performance of the 4-fold cross-validation is overly optimistic, so it can be comparable to the performance of the model when it is tested on data from a data source that was used during the training and validation phases. In contrast, the leave-source-out cross-validation and the test results indicate the performance of the model when it is tested on data from an unseen data source that was not used during training and validation. Hence, from the results by Leinonen et al. [16], it can be seen that the performance of the ResNet model drops considerably when tested on data from sources that were not used during the training and validation phases. For example, when the CPSC and CPSC-Extra dataset is used as the test set, the model's macro-AUROC is 0.89, but when the 4-fold cross-validation is used, the value is 0.97, showing a huge generalization gap when testing the model on a new data source.

Findings from Other Multi-Source Studies

There are some other studies that have made similar observations about poor generalization to new sources in ECG classification. For example, Han et al. [35], similar to Leinonen et al. [16], used multi-source data and leave-source-out cross-validation to assess the generalization performance of a deep learning model for 12-lead ECG classification. For performance evaluation, they used the so-called Challenge Metric, which is a task-specific evaluation metric introduced in the PhysioNet/Computing in Cardiology Challenge 2021 [36]. This metric gives full credit for correct diagnoses, while for clinically similar misdiagnoses, it gives partial credits. It is calculated using a weighted confusion matrix $A = [a_{ij}]$, in which the clinical similarity across classes is encoded by a domain-specific matrix $W = [w_{ij}]$ that weights each entry [36]. The overall unnormalized score is:

$$s = \sum_{i,j} w_{ij} \times a_{ij} \tag{2.1}$$

This score is standardized to range from 0 to 1, where 1 denotes perfect classification and 0 corresponds to a simple classifier that always predicts the "normal" class [36].

Even though the resulting value of the Challenge Metric of their proposed deep learning model, tested on a new source, was higher than that of a baseline model ($0.483 > 0.384$), it was still very low. Thus, this study also showed huge generalizability problems of deep learning models in ECG classification, despite using multi-source data.

Similarly, A. Ballas and C. Diou [37] also observed significant performance degradation when deep learning models trained on certain hospital datasets were tested on unseen hospital data. In their study, A. Ballas and C. Diou [37] demonstrated the distributional shift found in the ECG database, which comes from different sources, by experimentally showing the performance decline of trained models when evaluated on data from unseen domains. As expected, their model yielded a weaker performance on an unseen domain than on a seen one. For instance, when predicting the presence of left axis deviation, the model had an F1-score value of 0.79 when tested on a seen domain compared to a value of only 0.40 when tested on an unseen domain.

Need for Further Domain Generalization Research

Thus, as supported by the results of Leinonen et al. [16], Han et al. [35], and A. Ballas and C. Diou [37], it is important to explore, implement, and test some methods for enhancing the domain generalization of deep learning models. Because of its superior results and code availability, the ResNet model of Leinonen et al. [16] is used as the baseline model for this thesis, and multi-source domain-adversarial

training (DANN) and MixStyle are implemented in that model. The thesis uses the same data that was used by Leinonen et al. [16] due to its availability and relevance to the aims of the thesis.

2.2 Overview of Studies That Used Domain Generalization Methods

This subchapter presents related work on domain generalization in ECG classification with an emphasis on the two methods assessed in this thesis: multi-source domain-adversarial training (DANN) and MixStyle.

Selection and Rationale for Methods Assessed in This Thesis

These two methods were selected because they represent two fundamentally distinct and complementary approaches to domain generalization: multi-source domain-adversarial training (DANN) uses gradient-based learning to align feature distributions across domains [25], whereas the MixStyle technique uses a data augmentation mechanism to simulate domain shift directly in the feature space [26]. Both these approaches have demonstrated promising results in biomedical and image domains. Moreover, their compatibility with existing architectures, code availability, and ease of use make them appropriate starting points for implementation in the context of ECG classification. Even though there are many alternative domain generalization techniques, this study focuses on multi-source domain-adversarial training (DANN) and MixStyle as representative baselines for two common methodological paradigms.

Literature Search Strategy and Scope

In order to find relevant studies, a literature search was carried out in May 2024 using Google Scholar, IEEE Xplore, PubMed, and Web of Science. The following search

strings were used: “domain generalization” AND ECG, “domain adversarial neural networks” AND ECG, "domain shift" AND ECG classification, "cross-hospital generalization" AND ECG, and “MixStyle” AND ECG. The review was limited to peer-reviewed journal and conference papers published between 2017 and 2024. The selection of studies was based on their relevance to ECG classification across multiple data sources or hospitals using the two domain generalization methods: multi-source domain-adversarial training (DANN) and MixStyle.

Findings on Multi-Source Domain-Adversarial Training (DANN)

As of May 2024, this literature search returned only two highly relevant peer-reviewed studies focusing on multi-source domain-adversarial training (DANN) for ECG classification. This is consistent with what A. Ballas and C. Diou [37] said about the availability of studies with this focus. These two studies are by Hasani et al. [38] and Shang et al. [39]. They are explored in detail in this thesis (See Section 2.2.1).

Some other studies have been found on multi-source domain-adversarial training (DANN) that are only somewhat relevant to the thesis. Some of them are on the use of this technique on applications other than ECG. For instance, in 2021, Lin et al. [40] used multi-source domain-adversarial training (DANN) for domain generalization in the context of person re-identification. Some of the studies explored this technique on applications involving ECG, but not ECG classification. For example, in 2024, Wang et al. [41] used multi-source domain-adversarial training (DANN) to improve generalization in ECG-based cognitive load estimation. However, these studies are not highly relevant to the thesis since they do not explore multi-source domain-adversarial training (DANN) for domain generalization in the context of ECG classification, so they are not investigated in detail in this thesis. Furthermore, there are some studies that explore multi-source domain-adversarial training

(DANN) for domain adaptation in ECG classification, e.g., a study by Niu et al. [42]. However, these studies are not relevant to the thesis since domain adaptation is different from domain generalization.

Nevertheless, there are studies that explore the theoretical framework behind multi-source domain-adversarial training (DANN), and are useful for the thesis because of the theory and even source code that they provide. They include Y. Ganin and V. Lempitsky [43], Ganin et al. [25], M. H. Zonoozi and V. Seydi [44], Ajakan et al. [45], and Rangwani et al. [46]. These studies cover the entire theory behind multi-source domain-adversarial training (DANN) in great detail, so they are used to support the methodology of this technique (see Section 3.4.2).

Findings on MixStyle

Similarly, the literature search returned only one published study on using MixStyle for ECG generalization, conducted by Yang et al [47]. It is explored in detail in this thesis (See Section 2.2.2).

Some other studies have been found on MixStyle that are only somewhat relevant to the thesis. Some of them are on the use of this method on applications other than ECG. For example, in 2024, Xiao et al. [48] used MixStyle to improve domain generalization in the context of sound event detection. However, these studies are not highly relevant to the thesis since they do not explore the MixStyle method for domain generalization in the context of ECG classification, so they are not investigated in detail in this thesis.

Nevertheless, there is a paper by Zhou et al. [26] that originally introduces MixStyle for domain generalization. The study by Yang et al. [47] uses the MixStyle method based on this paper, which covers the entire theory behind MixStyle in great detail. Therefore, it is used to support the methodology of this technique (see Section 3.4.3).

2.2.1 Studies on Multi-Source Domain-Adversarial Training (DANN)

This subsection explores the two highly relevant studies on using multi-source domain-adversarial training (DANN) for ECG classification that were found from the literature search.

First Study - Hasani et al. (2020)

The **first study**, by Hasani et al. [38], aimed at designing a multi-source domain generalization model in order to solve the heartbeat classification problem and tackle the problem of distribution discrepancy that arises when data is gathered from multiple sources with different acquisition settings. The study was part of the PhysioNet/Computing in Cardiology Challenge 2020 [49]. The authors combined a convolutional neural network (CNN) and a long-short-term memory (LSTM) network and utilized them for feature extraction. Then, they integrated the adversarial domain generalization method into the model.

The study used the CPSC, CPSC-Extra [50], PTB-XL [51], and G12EC [30] datasets, which were also used by Leinonen et al. [16], to assess the effectiveness of the adversarial domain generalization method in ECG classification by treating the datasets as different domains and evaluating the model performance in a leave-one-out manner. Specifically, training was conducted on three domains, and the fourth domain was used as a test set, and then four different outcomes were averaged. As a result, the model without domain generalization achieved a Challenge Metric score of 0.343, while the domain generalization improved the performance of the model, which achieved a Challenge Metric score of 0.352 [38].

Second Study - Shang et al. (2021)

In the **second study**, by Shang et al. [39], the authors had a goal of training a domain-invariant model "to classify cardiac abnormalities from ECGs and evaluate the diagnostic potential of reduced-lead ECGs". The study was part of the PhysioNet/Computing in Cardiology Challenge 2021 [36]. In the study, the authors used a SE-ResNet model and implemented a domain-adversarial feature-learning scheme into the model. This domain-adversarial feature-learning scheme included a gradient reversal layer for the purpose of learning domain-invariant features.

The resulting domain-invariant model was trained on five datasets, including CPSC, CPSC-Extra [50], PTB [52], PTB-XL [51], and G12EC [30], and then locally validated on the unseen Ningbo [53] dataset. Apart from that, the baseline SE-ResNet model was trained on the same datasets in order to compare its performance with that of the modified model. As a result, when validating on the held-out dataset, the domain generalization method yielded a better performance than the baseline model. Table 2.1 shows the resulting Challenge Metric [36] scores for the baseline model (B) and the domain-invariant model (D) for different ECG leads. Both models were validated on data from a seen and an unseen domain. According to the table, it can be seen that the performances of both models experience a considerable drop when tested on the data from the unseen domain. However, the domain generalization model performs slightly better on the unseen domain than the baseline model, suggesting that this multi-source domain-adversarial training (DANN) may help enhance the model's generalizability [39]. In addition, from the table, it can be seen that the domain-adversarial method worsens the performance of the model when it is tested on data from a seen domain that was used during training.

Lead	12	6	4	3	2
B (seen domain)	0.75	0.71	0.73	0.73	0.71
D (seen domain)	0.72	0.68	0.69	0.69	0.68
B (unseen domain)	0.43	0.46	0.46	0.44	0.45
D (unseen domain)	0.44	0.49	0.48	0.48	0.49

Table 2.1: "Challenge Metric scores for the baseline and domain-invariant models for data in the seen (average 5-fold cross-validation of training data from CPSC, CPSC-Extra, PTB, PTB-XL, and G12EC) and unseen (Ningbo dataset) domains. B: baseline model. D: domain-invariant model" [39]. Created by Shang et al. [39]

Summary and Implementation Strategy

Overall, these two studies by Hasani et al. [38] and Shang et al. [39] show that multi-source domain-adversarial training (DANN) improves the generalization performance of deep learning models when they are tested on data from an unseen domain. Models trained with this technique exhibit increased robustness. However, the differences between the performances of the domain-adversarial generalization models and the baseline models are only slight. Both the domain-adversarial generalization model and the baseline model experience a significant performance drop when tested on data from unseen domains. This suggests that the implementation of multi-source domain-adversarial training (DANN) is only part of the deep learning generalizability improvement. There may be some additional challenges that need to be addressed to achieve robust generalization. Nevertheless, it is worth exploring these techniques and testing them using the model and the data from the study of Leinonen et al. [16]. Moreover, the thesis includes multiple experiments by changing hyperparameters, data portions, test sets, etc. This thorough experimentation with domain-adversarial generalization models was not done in the studies by Hasani et al. [38] and Shang et al. [39], and it has the potential to further improve the generalizability of deep learning models in ECG classification.

The thesis explores, builds upon, and implements the same methods that Hasani et al. [38] and Shang et al. [39] implemented. However, both Hasani et al. [38] and Shang et al. [39] do not provide source codes for their solutions in the articles. Nonetheless, after extensive research, some code on multi-source domain-adversarial training (DANN) has been found in an open-source and well-documented library for transfer learning on GitHub called "Transfer Learning Library". This code is taken and adapted for the ECG classification task of this thesis. The code [1] and the library [2] are available online.

2.2.2 Studies on MixStyle

This subsection explores the highly relevant study on using MixStyle for improving the domain generalizability of deep learning models in the context of ECG classification found from the literature search.

In 2021, Yang et al. [47] conducted a study to identify cardiac abnormalities from reduced-lead ECGs using MixStyle for domain generalization. The study was part of the PhysioNet/Computing in Cardiology Challenge 2021 [36]. The authors presented and used a Mixed-Domain self-Attention ResNet (MDARsn) model that combined a Squeeze-and-Excitation ResNet (SERsn) model architecture with MixStyle and Multi-Head Attention (MHA) layer in order to automatically learn the relationships between ECG leads and CVDs and make the model generalizable to new domains. MixStyle has shown promising results in domain generalization with its feature-based augmentation [47], so the authors decided to use that in their study.

There were five datasets representing five different domains, taken from the PhysioNet/Computing in Cardiology Challenge 2021 [36], that were used in the study. Three of these datasets were used for training, one for validation, and one for testing.

[1] `https://github.com/thuml/Transfer-Learning-Library/blob/master/examples/domain_adaptation/image_classification/dann.py`

[2] `https://github.com/thuml/Transfer-Learning-Library`

Model	Leads	Training	Validation	Test	Ranking
SERsn	12	0.721	0.582	-	-
	6	0.7	0.576	-	-
	4	0.704	0.580	-	-
	3	0.704	0.586	-	-
	2	0.699	0.580	-	-
SERsn+M	12	0.731	**0.602**	0.4	18
	6	0.709	**0.593**	0.33	23
	4	0.711	**0.597**	0.37	20
	3	0.713	**0.591**	0.34	23
	2	0.705	**0.589**	0.34	22
SERsn+M+A (**MDARsn**)	12	**0.738**	0.525	-	-
	6	**0.71**	0.506	-	-
	4	**0.723**	0.511	-	-
	3	**0.719**	0.503	-	-
	2	**0.707**	0.499	-	-

Table 2.2: "Challenge Metric scores for different models testing on the training set, repeated scoring on the hidden validation set, and one-time scoring on the hidden test set, as well as the ranking on the hidden test set. M: with MixStyle block; A: with MHA layer" [47]. Created by Yang et al. [47]

During the validation phase, several hyperparameters were optimized. The authors compared the performances of three models: a basic SERsn model, a SERsn model with MixStyle, and a SERsn model with MixStyle and MHA layer. The experiments were done for several ECG leads. Table 2.2 shows the results of the experiments, reporting the Challenge Metric [36] score on the training, hidden validation, and test sets of three models: SERsn, SERsn with MixStyle, and SERsn with MixStyle and an MHA layer (MDARsn model). According to the table, it can be seen that the SERsn with MixStyle yields a better performance across all ECG leads on the hidden validation set than both the basic SERsn and the MDARsn models.

Overall, the results of the study by Yang et al. [47] show that the MixStyle approach seems to have the potential to improve the generalizability of deep learning models for ECG classification. The model with MixStyle outperforms the baseline model when it is tested on the hidden validation set. However, when the model with

MixStyle is tested on the hidden test set, its performance drops and is similar to the performance of multi-source domain-adversarial training (DANN). This suggests that while MixStyle may contribute to improved generalization of deep learning models for ECG classification, it is not sufficient on its own. There may be some additional challenges that need to be addressed to achieve robust generalization. Nevertheless, it is worth exploring the MixStyle method and testing it using the model and the data from the study of Leinonen et al. [16]. Furthermore, the thesis conducts a few experiments involving MixStyle that Yang et al. [47] have not done, so it has the potential to further improve the generalizability of deep learning models in ECG classification.

The study by Yang et al. [47], like the studies by Hasani et al. [38] and Shang et al. [39], does not provide a source code for their solution. However, fortunately, Zhou et al. [26] provide a source code for their study on MixStyle for domain generalization, and it is taken and adapted for the ECG classification task of this thesis. The code is available online. [3]

[3] `https://github.com/thuml/Transfer-Learning-Library/blob/master/examples/domain_generalization/image_classification/mixstyle.py`

3 Methodology

This chapter presents a detailed description of all the methods that have been used in this thesis, including a detailed description of all the datasets used for the analysis. Moreover, this chapter includes the description and formulation of the baseline deep learning model, the description of multi-source domain-adversarial training (DANN) and MixStyle, and the description of the data preprocessing step. Finally, there are detailed descriptions of the model evaluation step and the experimental setup used in the study.

3.1 Data Description

There are two publicly available datasets used in this study: data provided by Shandong Provincial Hospital [54] and data from the George B. Moody PhysioNet Challenge 2021 [36]. In their study, Leinonen et al. [16] refer to them as SPH data and CinC data. The CinC data contains data from four different data sources, namely the Chapman-Shaoxing and Ningbo dataset, the CPSC and CPSC-Extra dataset, the G12EC dataset, and the PTB and PTB-XL dataset. The SPH data contains data from only one data source, which is the SPH dataset. Overall, there are five distinct data sources used in this study, and this multi-source data aligns with the main aim of the thesis, which is to improve the cross-hospital generalizability of deep learning models used for electrocardiogram (ECG) classification.

Table 3.1 provides a detailed overview of the study's five data sources. From

Sources	Countries	Locations	Total patients (n)	Total ECGs (n)	Included ECGs (n)	Sampling frequency	Length (s)	Labeling standard
Chapman-Shaoxing and Ningbo	China	Shaoxing People's Hospital Ningbo First Hospital	45,152	45,152	43,814	500	10	SNOMED
CPSC and CPSC-Extra	China	11 unnamed hospitals	Unknown	10,330	6,110	500	6-144	SNOMED
G12EC	USA	Emory University Hospital	15,738	10,344	8,892	500	5-10	SNOMED
PTB and PTB-XL	Germany and other European countries	University Clinic Benjamin Franklin Physkalisch Technische Bundesantalt	19,147	22,353	21,348	500, 1000	10-120	SNOMED
SPH	China	Shandong Provincial Hospital	24,666	25,770	23,274	500	10-60	AHA

Table 3.1: An overview of the study's five data sources, created by Leinonen et al. [16]

the table, it can be seen that the five data sources comprise data from various hospitals in different countries, including China, the USA, Germany, and some other European countries. Overall, the entire dataset consists of more than 113,000 12-lead ECGs. However, some of the recordings were excluded from the data, so the final dataset consists of 103,438 12-lead ECGs. Namely, the Chapman-Shaoxing and Ningbo dataset contains 43,814 recordings, the CPSC and CPSC-Extra dataset contains 6,110 recordings, the G12EC dataset contains 8,892 recordings, the PTB and PTB-XL dataset contains 21,348 recordings, and the SPH dataset contains 23,274 recordings.

3.1.1 Data from the George B. Moody PhysioNet Challenge 2021

The CinC data, which was taken from the George B. Moody PhysioNet Challenge 2021 [36], consists of 80,164 12-lead ECGs from multiple databases after the exclusion of some labels that are not of interest in this study. As mentioned earlier, the CinC data consists of several different data sources.

The **first data source**, selected by Leinonen et al. [16], corresponds to data from the China Physiological Signal Challenge (CPSC) [50], which was held in China in 2018. It contains data from patients that mainly come from China [16]. The data

source consists of two separate databases merged into one: the CPSC database and the CPSC-Extra database. The CPSC-Extra database contains data that was not used during the challenge [50]. Leinonen et al. [16] treat these two databases as one data source, naming it *CPSC and CPSC-Extra*. Overall, the CPSC and CPSC-Extra data source contains data collected from 11 unnamed hospitals in China [16]. Each 12-lead ECG recording lasted for a duration between 6 seconds and 60 seconds and was sampled at 500 Hz [50].

The **second data source** is the Physikalisch-Technische Bundesanstalt (PTB), which contains data from patients that mainly come from Germany and some other European countries [16]. It also consists of two separate databases, namely the PTB diagnostic ECG database [52] and the PTB-XL ECG dataset [51]. Just like CPSC and CPSC-Extra, Leinonen et al. [16] treat the PTB diagnostic ECG database and the PTB-XL ECG dataset as one data source, naming it *PTB and PTB-XL*. Overall, this data source comprises data that comes from two locations: University Clinic Benjamin Franklin and Physkalisch Technische Bundesantalt [16]. Both of these hospitals are located in Germany [51], [52]. The 12-lead ECG recordings from the original PTB database were sampled as 1,000 Hz [52], while the recordings from the PTB-XL dataset were sampled at 500 Hz [51]. Each 12-lead ECG recording from both databases lasted for a duration between 10 seconds and 120 seconds [36].

The **third data source** is the Georgia 12-lead ECG Challenge database (G12EC) [30], which contains data from patients that mainly come from the USA [16]. Leinonen et al. [16] refer to it as *G12EC*. The data source comprises data that comes from the Emory University Hospital, located in the USA [16]. Each 12-lead ECG recording lasted for a duration between 5 seconds and 10 seconds and was sampled at 500 Hz [36].

The **final data source of the CinC data** is referred to as *Chapman-Shaoxing and Ningbo* by Leinonen et al. [16], and it contains data from patients that mainly

come from China. The data source combines data from Chapman University, the Shaoxing People's Hospital database [55], and the Ningbo First Hospital database [53]. It comprises data that comes from two Chinese hospitals: Shaoxing People's Hospital and Ningbo First Hospital [16]. Each 12-lead ECG recording lasted for a duration of 10 seconds and was sampled at 500 Hz [53], [55].

The CinC data also contained the St. Petersburg INCART 12-lead arrhythmias database [56], which Leinonen et al. [16] decided to remove from the study because of significant differences in duration and size from the other sources. More details on this are contained in the work of Leinonen et al. [16].

In addition to ECGs, most of the recordings contain some demographic information, such as age and gender [16]. Table 3.2 illustrates this demographic information for each data source. From the table, it can be seen that the average age of patients from all the data sources used in the study is 57.05, which corresponds to middle-aged adults. Moreover, according to the table, in each data source, there are more male than female patients, with an average ratio of 56/44. Also, according to Table 3.1, some of the patients might have multiple ECG recordings. More details on this are contained in the work of Leinonen et al. [16].

3.1.2 Data from Shandong Provincial Hospital

The SPH database corresponds to a database that was collected between 2019 and 2020 at Shandong Provincial Hospital in China [54]. In addition to the four sources that make up the PhysioNet/CinC challenge 2021 data, Leinonen et al. [16] consider it as an independent data source, referring to it as *SPH*. This data source contains data from patients who mainly come from China. Each 12-lead ECG recording has a sampling frequency of 500 Hz and a duration between 10 seconds and 60 seconds [54].

Just like the recordings in the CinC data, the recordings in the SPH data, apart

Source	Age				Sex
	Mean (± SD)	Min	Max	U (n)	Ratio % (Male/Female)
Chapman-Shaoxing and Ningbo	58.21 (± 19.62)	0	89	44	56/43
CPSC and CPSC-Extra	62.69 (± 18.67)	1	104	8	57/43
G12EC	60.82 (± 15.51)	14	89	65	54/46
PTB and PTB-XL	59.57 (± 17.01)	2	95	92	52/48
SPH	49.66 (± 15.51)	18	95	0	57/43
Final dataset	57.05 (± 18.32)	0	104	209	56/44

U = Undefined | The Chapman-Shaoxing and Ningbo source has 14 ECGs that lack the sex value.

Table 3.2: "Demographic information per data source" [16], created by Leinonen et al. [16]

from ECGs, contain some additional information, including age and gender (Table 3.2). There were also a few patients with multiple recordings, and some identical and missing recordings. More details on this are contained in the work of Leinonen et al. [16].

3.2 Classification Labels

3.2.1 Harmonization: Mapping Between Different Labeling Standards

The CinC data and the SPH dataset follow different standards for ECG labeling. The CinC data uses Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) codes, while the SPH dataset is encoded using the American Heart Association (AHA) standard, as advised by the Heart Rhythm Society, the American College of Cardiology, and the American Heart Association. Since SNOMED CT codes are different from the AHA standard labeling, Leinonen et al. [16] de-

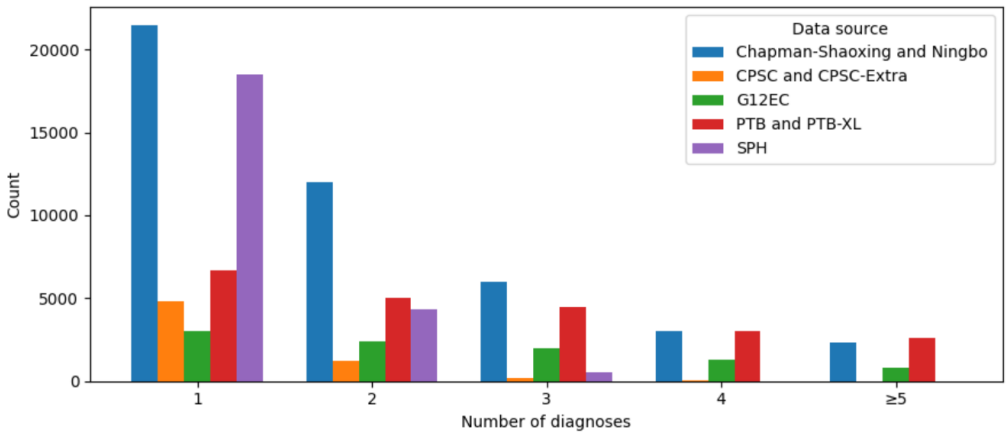

Figure 3.1: Number of patient diagnoses for each data source, adapted from Leinonen et al. [16]

veloped a label mapping in order to translate AHA codes to SNOMED CT. More details on this are contained in the work of Leinonen et al. [16].

3.2.2 Label Selection

Leinonen et al. [16] comprehensively selected labels for the study, with a total of 17 different cardiac diagnoses selected for ECG classification. As a result of this selection, the distributions of the diagnosis labels are different between each data source. Leinonen et al. [16] noted that this label imbalance is useful to the study since it reflects real-world conditions where data is collected from multiple hospitals, which differ from each other in terms of data distribution. The labels for these diagnoses were selected based on several criteria. More details on this are contained in the work of Leinonen et al. [16].

Figure 3.1 shows the number of patient diagnoses for each data source. From the figure, it can be seen that, although most ECG recordings have one, two, or three labels per patient, the number of labels per patient might vary from one to twelve, depending on the data source. A more detailed description of this is contained in the work of Leinonen et al. [16].

Establishing a mapping for one of the labels, the sinus rhythm label, between

AHA statements and SNOMED CT codes needed a different strategy because this label was not specifically included in AHA statements, but was included in SNOMED CT codes. More details on this are contained in the work of Leinonen et al. [16].

3.2.3 Description of Diagnoses

In total, in the study, there are 17 different labels that correspond to various cardiac diagnoses. Here are detailed descriptions of each diagnosis used in the study:

- **First Degree Atrioventricular (AV) Block** is a disease that affects the heart's electrical conduction system, causing electrical impulses to move more slowly than usual from the cardiac atria to the ventricles via the atrioventricular node (AV node). Although first-degree AV block usually does not produce any symptoms, it can lead to more serious heart block types like second and third-degree AV block. It corresponds to a PR interval greater than 200 milliseconds [57].

- **Atrial Fibrillation** is a type of cardiac arrhythmia that is characterized by the heart's atrial chambers pounding quickly and erratically [58]. Usually, it begins as brief episodes of abnormal beating that gradually lengthen or become continuous over time [59].

- **Atrial Flutter** is a type of cardiac arrhythmia that occurs when the heart's upper chambers, or atria, pump extremely quickly due to a short circuit [60]. Most of the time, it is characterized by an abnormal, irregular heart rhythm that appears suddenly [61].

- **Right Bundle Branch Block** is an obstruction in the right bundle branch that causes the heartbeat signal to be delayed and out of sync with the left bundle branch, resulting in an irregular heartbeat. In contrast to an incomplete

right bundle branch block, a complete one increases the risk of heart attack and death [62].

- **Incomplete Right Bundle Branch Block** is an obstruction in the right bundle branch that causes the heartbeat signal to be delayed and out of sync with the left bundle branch, resulting in an irregular heartbeat. In contrast to a complete right bundle branch block, an incomplete one does not increase the risk of heart attack and death [62].

- **Left Anterior Fascicular Block** is an abnormal cardiac condition when there is an interference with the signal from the heartbeat as it reaches the left anterior fascicle of the left bundle branch of the heart. As a result, the left ventricle of the heart contracts later than the right. It usually has no symptoms [63].

- **Left Axis Deviation** is an abnormal cardiac condition when the general direction of the electrical impulses as they travel through the heart shifts leftward beyond the normal range [64]. It can be caused by several potential factors. Symptoms and treatment of left axis deviation depend on the underlying factor [65].

- **Left Bundle Branch Block** is an obstruction of the electrical impulse that causes the heartbeat, resulting in an abnormal heart rhythm. During the left bundle branch block, there is either partial or complete blockage of the bundle branch that delivers the electrical impulse to the left ventricle, so the heart struggles to pump blood efficiently. If there are any symptoms, the left bundle branch block may worsen them and accelerate the heart's decline [66].

- **Low QRS Voltages** are defined by electrical signals seen on an electrocardiogram that are smaller than normal. This means that the peaks of the ECG

waves are smaller in height when the heart's electrical activity is less than normal [67]. Low QRS Voltages may come with various conditions, including obesity, massive myocardial infarction, etc [68].

- **Premature Atrial Contraction** is a type of cardiac arrhythmia that is characterized by extra heartbeats, which start in the upper chambers of the heart (atria). Usually, they have no symptoms, but sometimes they may feel like a shock in the chest or a skipped beat, which may cause the person to feel uneasy [69].

- **Right Axis Deviation** is an abnormal cardiac condition that occurs when the heart's electrical activity shifts to the right, possibly as a result of right bundle branch block or right ventricular hypertrophy [70]. It is asymptomatic by itself, and many of the symptoms exhibited by patients with Right Axis Deviation are linked to its different causes, including lateral myocardial infarction, right ventricular hypertrophy, etc [71].

- **Sinus Rhythm** is a cardiac rhythm where the sinus node marks the start of the cardiac muscle's depolarization [72]. It is required, but not sufficient, for the heart's normal electrical activity. The presence of P waves with a normal morphology on the electrocardiogram is indicative of a sinus rhythm [73]. In this case, Sinus Rhythm refers to the Normal Sinus Rhythm, which is generally regular and describes the distinctive rhythm of the healthy human heart [74].

- **Sinus Arrhythmia** is a type of normal sinus rhythm, where there is a normal heart rate variation, in which the heartbeat slightly speeds up and slows down in response to inhalations and exhalations. This type of arrhythmia is regarded as normal. Usually, it indicates that the heart is in good health [75].

- **Sinus Bradycardia** is a type of sinus rhythm, which is slower than usual (less than 60 beats per minute in an adult) but otherwise normal. Sometimes,

it can be a symptom of certain cardiac disorders or issues, but it can also be an indication that a person is in excellent health due to frequent activity [76]. When it is serious, it can result in symptoms such as lightheadedness, dizziness, hypotension, vertigo, and syncope [77].

- **Sinus Tachycardia** is a type of sinus rhythm when the heart rate is faster than 100 beats per minute [78]. Although sinus tachycardia is a common reaction to stressors like physical activity, when the heart rate increases to meet the body's higher demand for oxygen and energy, it can also be caused by a cardiac problem [79]. When it is not caused by stress, it can result in symptoms such as shortness of breath, fatigue, inability to handle exercise, and palpitations [78].

- **T Wave Abnormal** refers to an abnormality of a T wave, which is a component of an electrocardiogram that represents how the heart's ventricles repolarize (recover) after contracting [80]. In other words, the heart's recovery after a heartbeat. When a T wave is abnormal, it means that it does not look normal on an electrocardiogram. In other words, it might look flattened (-1.0 to 1.0 mm), inverted, symmetrical, biphasic, and peaked [81]. Although these abnormalities are thought to be quite harmless, physicians use them to direct treatment [82].

- **T Wave Inversion** refers to an inversion of a T wave that is deeper than 1.0 mm [81]. T wave inversion means that the T wave is flipped upside down, whereas the normal T wave points upwards. It is common among young, healthy people and athletes, but sometimes it can be a manifestation of cardiac problems, such as myocardial injury [83].

3.3 Data Preprocessing

Before training, validating, and testing the models on the datasets, some necessary data preprocessing was performed. The data preprocessing phase was carried out by following principles similar to those that were used in the study by Zhao et al. [24]. First, all the ECGs in the datasets were resampled to 250 Hz, and each recording was adjusted to 4096 time samples (or time points), or around 16 seconds, in duration. Subsequently, signals longer than 16 seconds were cropped at random. Shorter signals were extended to the required sample size by applying zero padding (adding extra zero-valued rows and columns to the input data [84]) to both sides. The proportion of padding between the sides was selected at random. Then, the range [0,1] was used to normalize the signal amplitudes. Finally, sex values were encoded using one-hot encoding, whereas age values were scaled to the interval [0,1] [16].

3.4 Models

3.4.1 Baseline Model

Overview of the SE-ResNet Baseline Model

The first model used in this study is a baseline model, which is a 12-lead ECG classification model that was first developed for the George B. Moody PhysioNet/CinC challenge 2020 by Zhao et al. [24]. The model utilizes an SE-ResNet architecture to detect various cardiac diagnoses from 12-lead ECG recordings. Figure 3.2 includes a diagram that illustrates the architecture of this model.

From the diagram of the model architecture, it can be seen that the main input to the model is a 12-lead ECG signal of the length of 4096 samples. There is also a secondary input that comprises the age and gender features represented as a vector

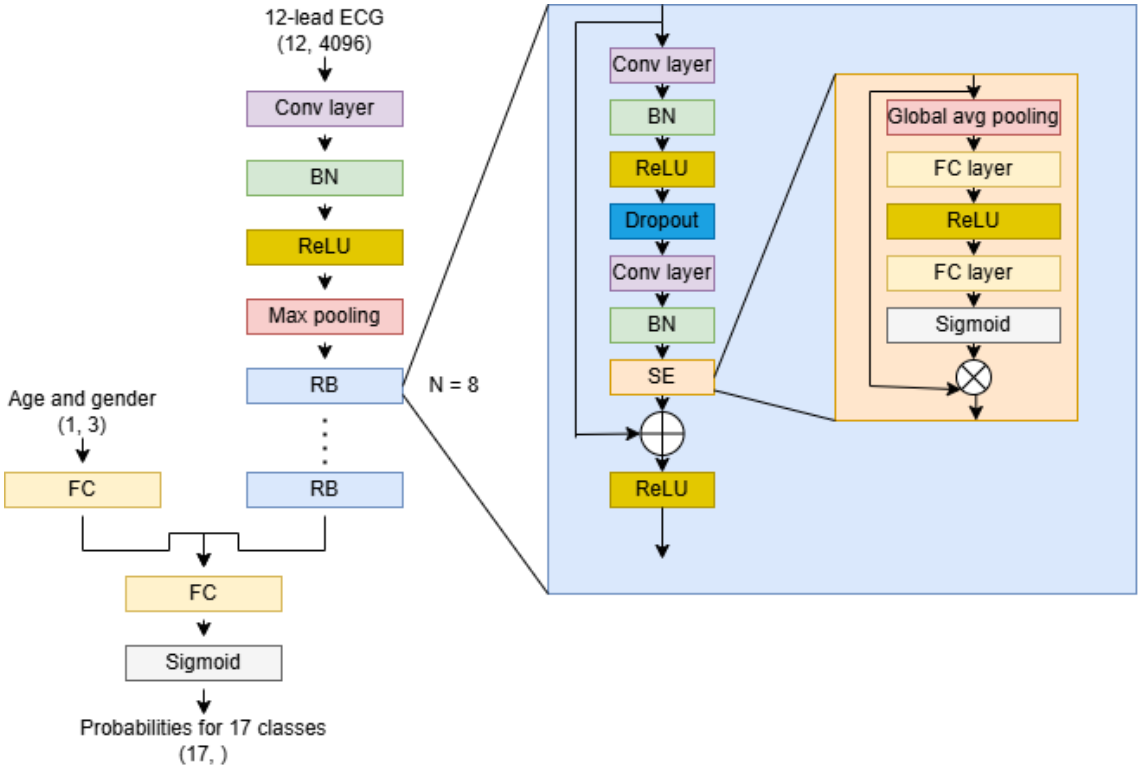

Figure 3.2: The architecture of the baseline SE-ResNet model, adapted from Zhao et al. [24] and Leinonen et al. [16]

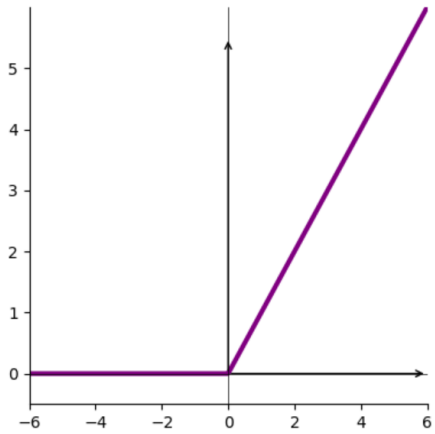

Figure 3.3: Illustration of the ReLU activation function

of shape (1,3).

Initial Layers

Before ECG classification, the ECG signal undergoes several preprocessing steps. First, it passes through a single convolutional layer. This layer is responsible for applying a convolution operation to the ECG signal input [85]. The convolution operation involves calculating the dot product between the values in the kernel and the input at each place after sliding a small window, also known as a kernel or filter, across the input data. This process produces a feature map that represents the identified features in the input [86]. In the context of the ECG classification task, the convolutional layer extracts local features from the unprocessed ECG signal.

Then, the extracted features pass through a batch normalization (BN) layer, which is used to normalize the outputs from the previous layer before transferring them as the input to the subsequent layer [87]. This is done to make the training process faster and more stable, via re-centering and re-scaling [88].

The next step involves an activation function that is used to introduce non-linearity to the network for improved feature learning. The SE-ResNet model in this study uses the rectified linear unit (ReLU) activation function that uses a simple

threshold operation, with negative values producing a zero output. As a result, all positive values remain constant, and all negative values are set to zero in the convolutional operation's output. The ReLU activation function is defined by the following formula: ReLU = $max(0, x)$ [84], [85]. Figure 3.3 illustrates the ReLU activation function.

After that, the input features pass through a pooling layer, which is used to downsample their spatial dimensionality and reduce their computational complexity. The SE-ResNet model uses a max pooling layer, where a window is slid across the feature map input. For each window, the greatest value is chosen, and therefore the most prominent feature in each local region is kept [84]. So, in other words, in the max pooling layer, the spatial dimensionality of the input features is reduced by retaining the most prominent ones.

Residual Blocks with Squeeze-and-Excitation (SE) Modules

Then, there are 8 residual blocks in the model that are used to enhance the extraction of features and prevent vanishing gradients [89]. The vanishing gradient problem is the problem of significantly different gradient magnitudes between earlier and later layers that arises when neural networks are trained by backpropagation. These techniques update the weights of neural networks in proportion to their partial derivative of the loss function [90]. Residual blocks are the fundamental building blocks of the ResNet (Residual Network) architecture.

In the SE-ResNet model architecture of the study, each residual block consists of two convolutional layers for additional deep ECG feature extraction, a batch normalization (BN) layer, a ReLU activation function, a dropout layer, and a Squeeze-and-Excitation (SE) block. The dropout layer is used to prevent overfitting by randomly deactivating neurons during training [91]. In this model, the dropout layer has a rate of 0.2 [24]. The squeeze-and-excitation (SE) block is used to suppress less in-

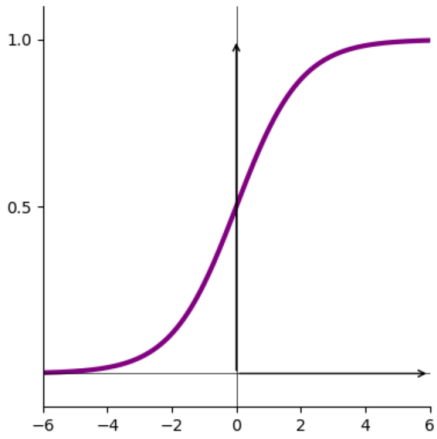

Figure 3.4: Illustration of the sigmoid activation function

formative feature channels while highlighting the more important ones [92]. In the context of the study, the SE block is able to give varying degrees of importance to individual leads, depending on how relevant they are to a particular cardiac diagnosis. The overall purpose of the SE block in this study is "to model spatial relationships between the ECG channels" [24].

The SE block itself has four constituents: a global average pooling layer, two fully connected (FC) layers, a ReLU activation function, a sigmoid activation function, and a multiplication operation. The squeezing operation of the block involves the global average pooling layer, which is used to capture channel-wise statistics so that each channel is squeezed to a single value. The excitation operation, which uses fully connected (FC) layers to describe the interdependencies across channels in the form of importance weights, receives the output of the squeezing operations. The ReLU activation function is again used to introduce non-linearity, whereas the sigmoid activation is used to scale each channel with the importance weights. The sigmoid activation function is an S-shaped function that outputs values in the range of 0 to 1 (see Figure 3.4). The multiplication operation is used in a way so that the learned weights are applied to the feature maps [92].

Auxiliary Inputs and Final Layers

The vector containing the age and gender feature input is processed in a fully connected (FC) layer and then concatenated with the ECG features, which are extracted from the residual blocks (RBs). This concatenated feature vector then passes through another fully connected layer, where the features are assigned to the output classes. Then, the feature vector passes through a sigmoid activation function, which is used to produce probability scores for the 17 different labels selected for ECG classification. The final output is then a vector of probabilities with a shape (17,) that shows the likelihood of each cardiac diagnosis from the 17 diagnoses selected for the study.

According to Leinonen et al. [16], all the convolutional layers in the model, including the first one and those inside the residual blocks, use 64 convolutional filters. With every additional RB unit, the number of filters doubles. Following the max pooling layer and the third, fifth, and seventh RBs, the feature dimension is cut in half. The first convolutional kernel is given a kernel size of 15, whereas the remaining kernels have a size of 7 [24].

Training Setup and Hyperparameters

Initially, the model had a batch size of 64 and was trained across 50 epochs. The loss function was selected to be binary cross-entropy loss, and an initial learning rate of 0.003 for the Adam optimizer was employed.

The binary cross-entropy loss is given by the following formula [93]:

$$L = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right] \tag{3.1}$$

where y_i is the ground truth label and p_i is the predicted probability for an ECG diagnosis.

Model Reference Name

In this study, this model is referred to as *SE-ResNet Baseline*.

3.4.2 Model with Multi-Source Domain-Adversarial Training (DANN)

Overview of SE-ResNet DANN Model

The second model used in this study is the baseline SE-ResNet model combined with multi-source domain-adversarial training (DANN). Since this thesis aims to improve the generalizability of the ECG classification model to new, unseen domains, the Se-ResNet model in the study utilizes adversarial multi-source domain generalization, which is based on domain-adversarial neural networks (DANN) by Ganin et al. [25].

The adversarial multi-source domain generalization method should be implemented in such a way that the final classification decisions are based on representations that are both discriminative for the primary goal (classifying cardiac diagnoses) and invariant to domain changes [39]. To achieve this, Ganin et al. [43] proposed the architecture of the domain-adversarial neural network for adversarial multi-source domain generalization. Figure 3.5 provides an illustration of this architecture.

DANN Architecture Components

As can be seen from the figure, the architecture of the domain-adversarial neural network for adversarial multi-source domain generalization consists of three main parts: a feature extractor, a label predictor, and a domain classifier.

The feature extractor and the label predictor are already present in the baseline SE-ResNet model and were described in Section 3.4.1. The purpose of the domain classifier is to identify the domain of origin for the data based on the extracted features by learning domain features from them [43]. So, as input, the model takes

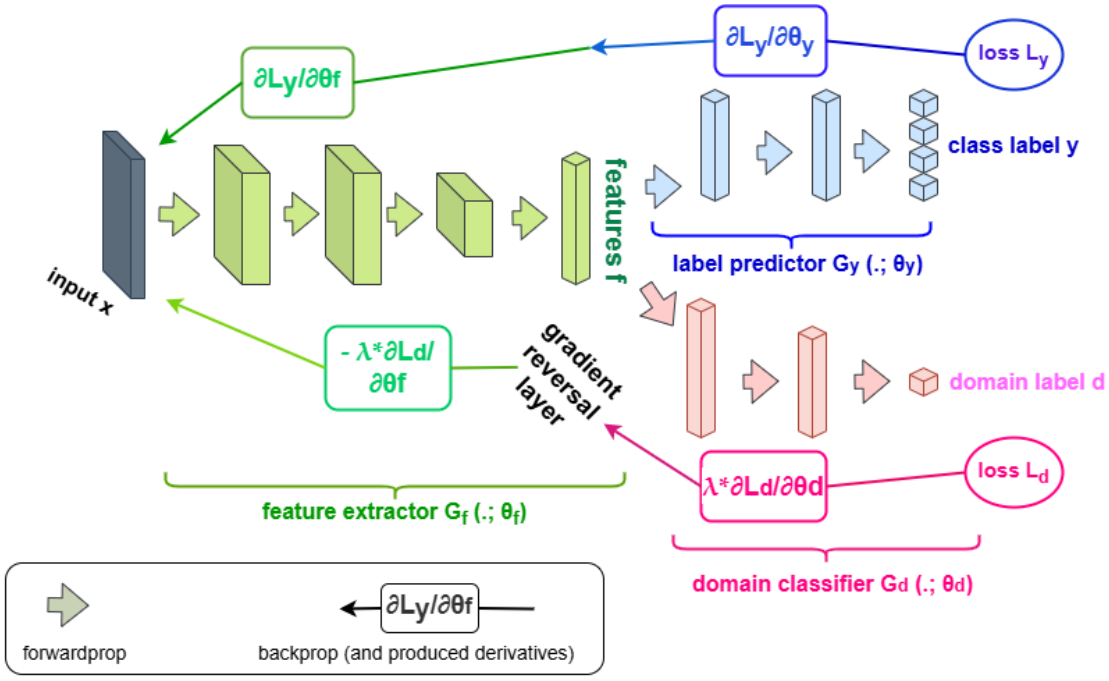

Figure 3.5: The architecture of the domain-adversarial neural network for adversarial multi-source domain generalization, adapted from Ganin et al. [25]

ECG signal data, which comes from multiple domains (corresponding to the different data sources described in Section 3.1. Then, this data proceeds to the feature extractor (denoted as G_f in Figure 3.5), which consists of the SE-ResNet model described in Section 3.4.1. The feature extractor processes the ECG signal data and extracts deep feature representations f. These feature representations subsequently proceed to both the label predictor G_y and the domain classifier G_d.

The label predictor G_y uses the features f for ECG classification by producing a class label y denoting different cardiac diagnoses. Also, the label predictor computes the classification loss L_y, which is then used for optimizing the model by using backpropagation [43]. Backpropagation is a gradient estimation method that calculates a loss function's gradient in relation to the network's weights using the chain rule [94]. So, using the loss L_y, the ability of the label predictor G_y to detect ECG diagnoses iteratively improves. The domain classifier G_d uses the features f

for domain discrimination by producing a domain label d denoting the prediction of which domain the sample comes from. The classifier also produces the domain loss L_d for optimizing the model. So, using the loss L_d, the ability of the domain classifier G_d to classify domains based on the features iteratively improves [43].

Gradient Reversal Layer and Loss Functions

However, during backpropagation, there is a layer inserted between the feature extractor G_f and the domain classifier G_d, which is called the gradient reversal layer (GRL). This layer flips the sign of the gradient, so that the partial derivatives $\frac{\partial L_d}{\partial \theta_f}$ get replaced by $-\frac{\partial L_d}{\partial \theta_f}$. Thereby, this gradient reversal forces the feature extractor to maintain class-discriminative features while eliminating domain-specific information. Thus, domain-invariant features are produced because the gradient reversal layer makes it hard for the domain classifier to distinguish between different domains [43].

When it comes to the domain loss L_d, during backpropagation, the domain classifier tries to minimize it, whereas the feature extractor tries to maximize it, while minimizing the classification loss L_y, because the gradient for the loss is reversed in the GRL. This simultaneous maximization of the domain loss L_d and the minimization of the classification loss L_y by the feature extractor results in domain-invariant features. The label predictor also tries to minimize the classification loss L_y [43]. The domain loss L_d is weighted by the hyperparameter λ [39]. This hyperparameter serves as a trade-off between the classification loss L_y and the domain loss L_d [44]. During the validation phase, it is tweaked and tuned to see which value yields the best domain generalization ability of the model.

Overall, the total loss function is [38]:

$$L = L_y + \lambda L_d \tag{3.2}$$

The classification loss L_y uses the usual binary cross-entropy loss, given in Formula 3.1, while the domain loss L_d uses the cross-entropy loss for the domain classifier, given by the following formula:

$$L = -\sum_{i=1}^{n} d_i \log(\hat{d}_i) \tag{3.3}$$

where d_i is the true domain label and $\hat{d}_i$ is the predicted probability for domain i.

Optimization Challenges and Practical Considerations

The optimization of Domain-Adversarial Neural Networks (DANN) is nontrivial because of the adversarial component introduced by the gradient reversal layer (GRL), even if the loss function employed in DANN appears straightforward in formulation: a weighted sum of classification loss L_y and domain loss L_d. By inverting the domain classifier's gradient signal, the GRL encourages the feature extractor to learn domain-invariant features. Thus, a min-max dynamic is introduced between the feature extractor and the domain classifier, similar to what Generative Adversarial Networks (GANs) [43] exhibit.

Because of this, the optimization does not ensure convergence to a global minimum, and hyperparameters such as λ, learning rate, and architecture depth might affect stability. According to a number of studies, including Ganin et al. [25], training DANN models can be unstable or result in poor feature representations if the domain loss outweighs the task loss, or vice versa. Similarly, Ajakan et al. [45] stated that feature collapse may result from an inappropriate balance between the classification loss and domain loss.

Furthermore, according to recent studies, including a study by Rangwani et al. [46], domain adversarial training may converge to local minima that lead to poor generalization on the target domain, especially in highly imbalanced datasets, such

as the data in this study. In order to improve convergence and model performance, it has been suggested to carefully tune λ and suitably use batch normalization [25], [43]. This is addressed via hyperparameter tuning in the experiments of the study (see Section 3.6).

Domain Classifier Architecture

In the architecture of the model in the study, the domain classifier uses two fully connected (FC) layers with 100 hidden units and ReLU activation.

Model Reference Name

In this study, this model is referred to as *SE-ResNet DANN*.

3.4.3 Model with MixStyle

Overview of SE-ResNet MixStyle Model

The third model used in this study is the baseline SE-ResNet model combined with the MixStyle method. This method, proposed by Zhou et al. [26], has shown a generalization capability with feature-based augmentation [47]. This corresponds to the thesis's goal to improve the generalizability of the ECG classification to new, unseen domains, so the method is implemented in this study.

How MixStyle Achieves Domain Generalization

The MixStyle method improves the model's ability to generalize to new, unseen domains by mixing the feature statistics of training instances from several domains. With MixStyle, domain generalization is attained by perturbing the ECG signals' feature statistics, creating new feature distributions that broaden the range of training domains [26]. This is motivated by the finding that feature statistics (mean and

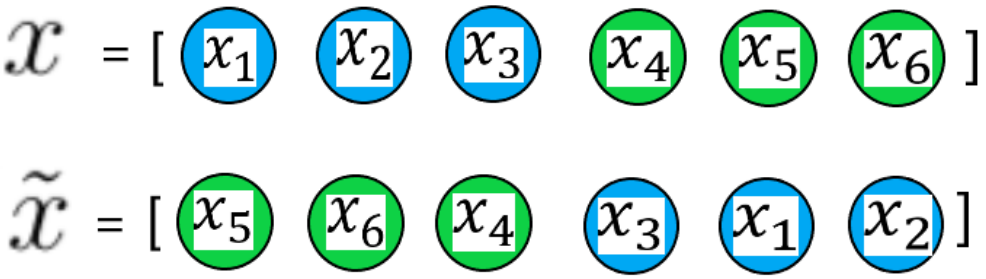

Figure 3.6: A visual representation of the generation of a reference batch when the domain labels are known. The domain label is denoted by color. Adapted from Zhou et al. [26]

standard deviation) rather than raw values frequently contain domain-specific variability in the data. MixStyle reduces the model's reliance on domain-specific biases by efficiently simulating unseen domains during training [47]. In the study, the domain labels are known, so the MixStyle method here is called cross-domain [26].

MixStyle Transformation Process

The MixStyle method works in the following way. First, when feature representations are extracted, their instance-level feature statistics, namely the mean and standard deviation, are computed. Then, given an input batch x of feature instances, a reference batch $\tilde{x}$ is created by shuffling the samples while maintaining domain separation. In the context of the thesis, the domain labels are given, so x is sampled from i and j, two distinct domains, as in $x = [x^i, x^j]$. The batch size of x^i and x^j is the same. Next, by switching the positions of x^i and x^j, $\tilde{x}$ is obtained. Each batch is then subjected to a shuffling operation along the batch dimension, so that $\tilde{x} = [\text{Shuffle}(x^j), \text{Shuffle}(x^i)]$ [26]. Figure 3.6 provides a graphical illustration of the process. After shuffling, the MixStyle transformation takes place, where a convex combination of the original and shuffled statistics is computed, ensuring a smooth interpolation between different domain distributions [26].

The MixStyle transformation is formulated as follows [26], [47]:

$$\beta_{\text{mix}} = \lambda\mu(x) + (1 - \lambda)\mu(\tilde{x}) \tag{3.4}$$

$$\gamma_{\text{mix}} = \lambda\sigma(x) + (1 - \lambda)\sigma(\tilde{x}) \tag{3.5}$$

where μ and σ are the mean and standard deviation of feature maps, and $\lambda \in \mathbb{R}^B$ are instance-wise random weights sampled from the Beta distribution, $\lambda \sim \text{Beta}(\alpha, \alpha)$ with $\alpha \in (0, \infty)$ being a hyperparameter [26]. This hyperparameter controls the influence of each domain's statistics and is tweaked and tuned to see which value yields the best domain generalization ability of the model.

Finally, a mixture of feature statistics is generated:

$$\text{MixStyle}(x, \tilde{x}) = \gamma_{\text{mix}} \times \frac{x - \mu(x)}{\sigma(x)} + \beta_{\text{mix}} \tag{3.6}$$

which corresponds to a domain-agnostic feature representation [26], [47].

Implementation Details

In the architecture of the model in the study, MixStyle is applied in the early convolutional layers of SE-ResNet, specifically in the first and second layers, since previous research has shown that deep neural networks' early layers are where domain-specific information is mostly found [47]. It is not applied in the later layers in order to maintain discriminative information for ECG classification. The model is trained using a standard binary cross-entropy loss, just like the baseline model.

Model Reference Name

In this study, this model is referred to as *SE-ResNet MixStyle*.

3.5 Model Performance Evaluation

Rationale for Metric Selection

For evaluating the performance of the models in the study, the area under the ROC curve (AUROC) was used. This metric is used to assess the model's capacity to differentiate across classes irrespective of the absolute counts of each class, which makes it suitable for the study's imbalanced dataset. Indeed, as Figure 3.1 in Section 3.2.2 shows, the dataset is quite imbalanced due to the highly uneven distribution of the diagnosis labels.

Understanding the ROC Curve

The concept behind the area under the ROC curve (AUROC) starts with the confusion matrix, which contains TPs, FPs, FNs, and TNs. TPs correspond to true positives, which denote samples where the model correctly predicted the positive class. On the other hand, FPs correspond to false positives, which denote samples where the model incorrectly predicted the positive class. Similarly, FNs correspond to true negatives, which denote samples where the model incorrectly predicted the negative class. Finally, TNs correspond to true negatives, which denote samples where the model correctly predicted the negative class [95]. In this case, the positive and negative classes simply denote the two possible outcomes that a model attempts to detect.

These four values, namely TP, FP, FN, and TN, can be used to calculate various measures for evaluating the model's performance, including true positive rate (TPR) and false positive rate (FPR). The true positive rate (also referred to as TPR or sensitivity) reflects the model's capacity to accurately identify samples that have the outcome of interest, such as sick individuals. Below is the formula for TPR [96]:

$$\begin{aligned} \text{TPR} &= \frac{\text{Number of samples, correctly predicted as having the positive class}}{\text{Number of actual samples with the positive class}} \\ &= \frac{\text{TP}}{\text{TP} + \text{FN}} \end{aligned} \tag{3.7}$$

The false positive rate (FPR) reflects the model's tendency to incorrectly classify negative samples as positive. Below is the formula for FPR [96]:

$$\begin{aligned} \text{FPR} &= \frac{\text{Number of samples, incorrectly predicted as having the positive class}}{\text{Number of actual samples with the negative class}} \\ &= \frac{\text{FP}}{\text{TN} + \text{FP}} \end{aligned} \tag{3.8}$$

The TPR and FPR metrics are used to build the so-called ROC space, which illustrates relative trade-offs between TP (benefits) and FP (costs). On the ROC space, the x-axis corresponds to the FPR, whereas the y-axis corresponds to the TPR. Each (FPR, TPR) pair is an operating point, which corresponds to a particular decision threshold (also known as the decision cut-off or classification threshold) [95]. Different (FPR, TPR) pairs are produced by these thresholds, which ultimately combine to form a continuous curve. This curve is called a ROC (receiver operating characteristic) curve and is used to illustrate the performance of a binary classifier [97]. Figure 3.7 illustrates an example of the ROC curve.

Area Under the ROC Curve (AUROC)

The ROC curve can be used to compute the area under the ROC curve (AUROC) metric, which is an expected classification performance metric that averages over all the decision thresholds [98]. The range of the values of AUROC is between 0 and 1 since it is calculated in the unit square. The larger the value of AUROC, the better the solution for the classification problem [99].

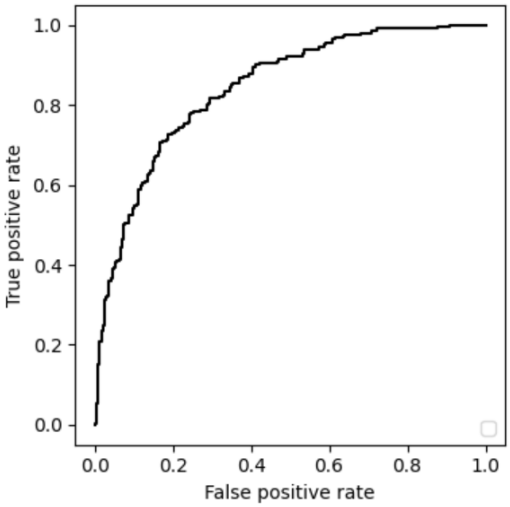

Figure 3.7: An example of the ROC curve, adapted from C. E. Metz [96]

AUROC in Multi-Label ECG Classification

The ECG classification task in this study is a multi-label task since there are 17 labels present in the data (see Section 3.2). Thus, in this case, the ROC curves are approached by using the one-vs-rest (OvR) perspective, in which, of labels c_i, the label c_i is treated as the positive class and all other labels are used as the negative class [97], where $i = 1, ..., n$ ($n = 17$ is the total number of labels).

Finally, using all the aforementioned information, the macro-averaged and micro-averaged AUROC metrics can be computed in the multi-label context of the study. While the micro-averaged AUROC characterizes a model's overall discriminative power across all classes [100], the macro-averaged AUROC provides insight into how effectively a model generalizes across different classes independently [101]. Macro-averaging is less sensitive to imbalanced class distributions within the data than micro-averaging since each class makes an equal contribution to the macro-averaged result. Therefore, as the study's dataset is quite imbalanced, only the macro-averaged AUROC is used here as a metric for model performance evaluation. By treating each label equally, the macro-averaged AUROC is calculated by averaging the total of the AUROCs for each label [16]. Here is the formula for the macro-

averaged AUROC:

$$\text{AUROC}_{\text{macro}} = \frac{1}{n} \sum_{i=1}^{n} \text{AUROC}_i \tag{3.9}$$

where AUROC_i is the AUROC value for the i-th label and n is the total number of labels ($n = 17$).

3.6 Description of Experiments

This section presents the experimental design used to evaluate and compare the performance of three deep learning models for ECG classification: the SE-ResNet Baseline model, the SE-ResNet DANN model, and the SE-ResNet MixStyle model. The aim is to investigate the impact of various domain generalization methods, as well as to assess how hyperparameter settings, dataset size, and test domain selection influence model generalizability.

Each experiment is designed to answer one or more research questions stated in Section 1.4. The **"Baseline vs. DANN with λ Tuning"**, **"Impact of Test Domain Selection"**, **"Tuning DANN for SPH Test Domain"**, and **"MixStyle vs. DANN and Baseline Comparison"** Experiments correspond to the **second research question**, which is: "To what extent do these domain generalization techniques improve the performance of deep learning models when evaluated on data from previously unseen sources?" The **"Baseline vs. DANN with λ Tuning"**, **"Effect of Batch Size on Generalization"**, **"Effect of Learning Rate on Generalization"**, **"Testing Extreme λ Values"**, **"Tuning DANN for SPH Test Domain"**, **"Generalization Under Data Scarcity"**, and **"MixStyle vs. DANN and Baseline Comparison"** Experiments correspond to the **third research question**, which is: "How do factors such as the amount of training data and hyperparameter settings influence the effectiveness of these generalization meth-

ods?"

There were many experiments designed and carried out in this study. As there were five domains (CPSC and CPSC-Extra, PTB and PTB-XL, G12EC, Chapman-Shaoxing and Ningbo, and SPH) used in the study, it was decided to use one of them for testing and the other four for training and validation. Initially, the performance of the SE-ResNet Baseline model was compared mainly to that of the SE-ResNet DANN model, since the DANN method is of greater importance to the study than the MixStyle method. The comparison with the SE-ResNet MixStyle model was used as one of the various experiments.

Since data from four domains was used for both training and validation, it was decided to use 75% of the data from each domain for training and 25% of the data from each domain for validation. Initially, it was decided to use CPSC and CPSC-Extra for testing, and the other four domains for training and validation.

It was also decided to perform data splits by using 25%, 50%, 75%, and 100% of the data from each domain for both training and validation in order to see the differences in the performance of the models across different data amounts and the effect of different amounts on the model performance and domain generalizability. Table 3.3 shows the resulting numbers of ECG recordings in each domain for training and validation across these splits. In the table, for simplicity, the PTB and PTB-XL domain is denoted as PTB, and the Chapman-Shaoxing and Ningbo source is denoted as Chapman.

In Section 3.4.1, it was mentioned that, initially, the batch size for the baseline model was 64, the learning rate was 0.003, and the number of epochs was 50. These hyperparameter values were also used to train the SE-ResNet DANN model and the SE-ResNet MixStyle model.

All the models in this study were trained and evaluated using CSC's supercomputer Puhti. The source code, containing the code for the three models, as well as

Different Data Splits			
25% of Data	50% of Data	75% of Data	100% of Data
TRAINING:			
G12EC: 1668 **SPH:** 4376 **PTB:** 4001 **Chapman:** 8211	**G12EC:** 3314 **SPH:** 8752 **PTB:** 8003 **Chapman:** 16422	**G12EC:** 5004 **SPH:** 13128 **PTB:** 12004 **Chapman:** 24632	**G12EC:** 6672 **SPH:** 17504 **PTB:** 16005 **Chapman:** 32843
VALIDATION:			
G12EC: 555 **SPH:** 1443 **PTB:** 1336 **Chapman:** 2743	**G12EC:** 1110 **SPH:** 2885 **PTB:** 2672 **Chapman:** 5486	**G12EC:** 1665 **SPH:** 4328 **PTB:** 4007 **Chapman:** 8228	**G12EC:** 2220 **SPH:** 5770 **PTB:** 5343 **Chapman:** 10971

Table 3.3: Number of ECG recordings in each domain used for training and validation across different data splits

the code for training and testing them, is freely available online on GitHub. [1]

The names and descriptions of each experiment are presented below.

3.6.1 Baseline vs. DANN with λ Tuning

The **main purpose of this experiment** is to determine whether domain-adversarial training using the DANN method improves generalization performance over the SE-ResNet Baseline model.

In this experiment, the performances of the SE-ResNet Baseline and the SE-ResNet DANN models were compared by testing them on the data from the CPSC and CPSC-Extra domain using the settings described earlier. The comparison was done across each data split. Also, hyperparameter selection was performed, and the optimal value of the hyperparameter λ of the SE-ResNet DANN model was selected using the validation set. Specifically, several SE-ResNet DANN models with different values of λ were compared in terms of the performance on the validation set, and the model with the best performance was chosen for testing and comparison with the

[1] `https://github.com/aituar17/Domain-Generalization-for-ECG-Classification`

SE-ResNet Baseline model. In their study, Shang et al. [39] empirically set 0.05 as the value for the hyperparameter λ, so it was used in this experiment. Additionally, 0.01, 0.1, 0.3, 0.5, and 1 were chosen as different values of λ for validation and hyperparameter selection. This range was chosen based on the range of values between 0.01 and 1, which was used by Ajakan et al. [45] in their study on DANN. For each data split (25%, 50%, 75%, and 100%), the optimal value of λ was chosen separately and independently.

3.6.2 Effect of Batch Size on Generalization

The **main purpose of this experiment** is to investigate how the choice of batch size affects the model's ability to generalize to unseen domains and to determine if larger or smaller batch sizes have an impact on cross-hospital performance in the context of domain-adversarial training.

In this experiment, for each data split, several SE-ResNet DANN models with different batch sizes were compared in terms of performance on the validation set, and the model with the best performance was selected for testing and comparison with the SE-ResNet Baseline model. In addition to 64, the batch sizes of 16, 32, 128, and 256 were chosen for this experiment. Furthermore, all these models were evaluated on the test set in order to analyze the effect of the batch size on cross-hospital generalization. This comparison was conducted solely for illustrative purposes and not for selecting the final model or tuning hyperparameters.

3.6.3 Effect of Learning Rate on Generalization

The **main purpose of this experiment** is to investigate how the choice of learning rate affects the model's ability to generalize to unseen domains and to determine if larger or smaller learning rates have an impact on cross-hospital performance in the context of domain-adversarial training.

In this experiment, for each data split, several SE-ResNet DANN models with different learning rates were compared in terms of the performance on the validation set, and the model with the best performance was selected for testing and comparison with the SE-ResNet Baseline model. In addition to 0.003, the learning rates of 0.00001, 0.0001, 0.001, 0.01, and 0.1 were chosen for this experiment. Furthermore, all these models were evaluated on the test set in order to analyze the effect of the learning rate on cross-hospital generalization. Similar to the "Effect of Batch Size on Generalization" Experiment, this comparison was done for illustrative purposes only.

3.6.4 Testing Extreme λ Values

The **main purpose of this experiment** is to observe the effect of extreme values of the hyperparameter λ on the model performance and domain generalization. This experiment aims to validate whether the previously recommended range of $\lambda \in [0.01, 1]$ holds in the ECG domain generalization context.

The extreme values chosen for this experiment were: 0.001, 5, 10, 50, 100, 500, and 1000. For each data split, similarly to the "Effect of Batch Size on Generalization" Experiment, these models were evaluated on the test set for illustrative purposes.

3.6.5 Impact of Test Domain Selection

The **main purpose of this experiment** is to assess the SE-ResNet DANN model's generalization and robustness when tested on different unseen domains and investigate whether the model's performance changes based on the domain selected as the test set.

In this experiment, instead of the CPSC and CPSC-Extra data source, the other four domains were separately used for testing. Additionally, the performance of the

SE-ResNet DANN model was compared to that of the SE-ResNet Baseline model using these different test sets. Different data splits for training and validation sets were done in a similar fashion to that of Table 3.3. In Appendix A, Table A.2 shows the number of ECG recordings in each domain for training and validation across different splits, when the G12EC domain is used as a test set. Similarly, Table A.3 shows the numbers when the SPH domain is used, Table A.4 shows the numbers when the PTB and PTB-XL domain is used, and Table A.5 shows the numbers when the Chapman-Shaoxing and Ningbo domain is used as a test set. In these tables, for simplicity, the CPSC and CPSC-Extra domain is denoted as CPSC. In this experiment, 0.05 was used as the value of λ for each data split and with each domain as a test set.

3.6.6 Tuning DANN for SPH Test Domain

The **main purpose of this experiment** is to investigate if the SE-ResNet DANN model's generalization performance may be enhanced even further on the SPH test domain.

In this experiment, as the SE-ResNet DANN model showed the best performance using the SPH domain as a test set (see Section 4.5), compared to the other domains, some additional experimentation was done on the model using SPH as a test set. In particular, for each data split, several SE-ResNet DANN models with different values of λ were compared in terms of the performance on the validation set, and the model with the best performance was chosen for testing and comparison with the SE-ResNet Baseline model. Similar to the "Baseline vs. DANN with λ Tuning" Experiment, 0.01, 0.05, 0.1, 0.3, 0.5, and 1 were chosen as different values of λ for validation.

3.6.7 Generalization Under Data Scarcity

The **main purpose of this experiment** is to investigate how extreme data scarcity affects model performance and domain generalization by training the SE-ResNet DANN and SE-ResNet Baseline models with only 5% and 10% of the available data. This experiment aims to examine whether domain generalization methods remain useful with very limited training data.

Different Data Splits	
5% of Data	10% of Data
TRAINING:	
G12EC: 334 **CPSC:** 229 **PTB:** 800 **Chapman:** 1642	**G12EC:** 667 **CPSC:** 458 **PTB:** 1600 **Chapman:** 3284
VALIDATION:	
G12EC: 111 **CPSC:** 76 **PTB:** 267 **Chapman:** 549	**G12EC:** 222 **CPSC:** 153 **PTB:** 534 **Chapman:** 1097

Table 3.4: Number of ECG recordings in each domain used for training and validation across 5% and 10% data splits, when the SPH domain is used as a test set

In this experiment, in addition to 25%, 50%, 75%, and 100%, the SE-ResNet DANN and SE-ResNet Baseline models were trained using 5% and 10% of the data from each domain for both training and validation. It was performed across different domains used as test sets. A fixed value of 0.05 for λ was used for training. Table 3.4 shows the resulting numbers of ECG recordings in each domain for training and validation across the 5% and 10% splits when the SPH domain is used as a test set. In Appendix A, Tables A.6, A.7, A.8, and A.9 show the numbers when the G12EC, the CPSC and CPSC-Extra, the PTB and PTB-XL, and the Chapman-Shaoxing and Ningbo domains are used as test sets, respectively.

3.6.8 MixStyle vs. DANN and Baseline Comparison

The **main purpose of this experiment** is to determine whether the MixStyle method improves generalization performance over the SE-ResNet Baseline and SE-ResNet DANN models.

In this experiment, the SE-ResNet MixStyle model was trained, validated, and tested, and its performance was compared to those of the SE-ResNet DANN and SE-ResNet Baseline models. The comparison was done across each data split (25%, 50%, 75%, and 100%) and using the SPH domain as a test set (since Section 4.5 showed that the SE-ResNet DANN model yields the best performance when the SPH domain is used as a test set). Furthermore, hyperparameter selection was performed. In other words, several SE-ResNet MixStyle models with different values of the hyperparameter α were compared in terms of the performance on the validation set, and the model with the best performance was chosen for testing and comparison with the SE-ResNet DANN and SE-ResNet Baseline models. In their studies, Yang et al. [47] and Zhou et al. [26] set 0.1 as the value for α, so it was used in this experiment. Additionally, 0.2, 0.3, 0.5, and 1 were chosen as different values of α for validation and hyperparameter selection. For each data split, the optimal value of α was chosen separately and independently.

During the comparison of the performances of models with different values of various hyperparameters, in the case of equal values of the macro-averaged AUROC, the optimal model was selected based on the values of other metrics, e.g., the macro-averaged precision, the challenge metric, etc.

4 Results

This chapter presents the results of all the experiments stated in Section 3.6 and some key observations.

4.1 Results of the "Baseline vs. DANN with λ Tuning" Experiment

	25%	50%	75%	100%
SE-ResNet DANN	0.92	0.90	0.91	0.92
SE-ResNet Baseline	0.91	0.89	0.91	0.90

Table 4.1: Testing results of the "Baseline vs. DANN with λ Tuning" Experiment across different data splits (macro-averaged AUROC), using $\lambda = 0.05$ for training

Table 4.1 shows the resulting values of the macro-averaged AUROC of the "Baseline vs. DANN with λ Tuning" Experiment, where both the SE-ResNet DANN and SE-ResNet Baseline models were tested on the CPSC and CPSC-Extra domain across different data splits (25%, 50%, 75%, 100%). From the table, it can be seen that, with $\lambda = 0.05$ used in the training, the SE-ResNet DANN model outperformed the SE-ResNet Baseline model in all of the cases, except for 75% of the data, where the performances of both models are equal. With 25% and 50% of the data, the SE-ResNet DANN model outperformed the SE-ResNet Baseline model by 0.01, whereas, with 100% of the data, the difference is 0.02.

	25%	50%	75%	100%
SE-ResNet DANN	0.91 ($\lambda = 0.01$)	0.89 ($\lambda = 1$)	0.91 ($\lambda = 0.05$)	0.91 ($\lambda = 1$)
SE-ResNet Baseline	0.91	0.89	0.91	0.90

Table 4.2: Testing results of the "Baseline vs. DANN with λ Tuning" Experiment across different data splits (macro-averaged AUROC), after selecting the optimal value of λ for each data split

Then, the optimal value of λ was selected for each data split using the validation set. Table 4.2 shows the resulting test values of the macro-averaged AUROC after performing the hyperparameter selection. From the table, it can be seen that the values are the same between both models across all data splits, except 100%, where the macro-averaged AUROC value of the SE-ResNet DANN model is greater than that of the SE-ResNet Baseline model ($0.91 > 0.90$). Overall, the performance of the SE-ResNet DANN model is worse after doing hyperparameter selection with the validation set, and the model originally trained with $\lambda = 0.05$ (as suggested by Shang et al. [39]) yields a better performance and outperforms the baseline model in most cases. Apart from the metric values, the table also shows the resulting values of the hyperparameter λ of the SE-ResNet DANN model chosen during validation for each data split. According to the table, the selected values are the same between 50% and 100% ($\lambda = 1$), but different otherwise.

Furthermore, from Tables 4.1 and 4.2, it can be seen that there is no trend in the macro-averaged AUROC values across different data sizes for both models. In other words, the value of the macro-averaged AUROC does not seem to vary between different data sizes (25%, 50%, 75%, and 100%).

Table 4.3 shows the validation results of the "Baseline vs. DANN with λ Tuning" Experiment across different data splits and models, such as the baseline model and SE-ResNet models with different values of λ. This table was used for the selection of the optimal value of λ for each data split. From the table, it can be seen that the

	25%	50%	75%	100%
$\lambda = 0.01$	0.96	0.96	0.97	0.97
$\lambda = 0.05$	0.96	0.96	0.97	0.96
$\lambda = 0.1$	0.96	0.96	0.97	0.97
$\lambda = 0.3$	0.95	0.96	0.97	0.96
$\lambda = 0.5$	0.96	0.96	0.97	0.97
$\lambda = 1$	0.96	0.97	0.96	0.97
SE-ResNet Baseline	0.96	0.97	0.97	0.97

Table 4.3: Validation results of the "Baseline vs. DANN with λ Tuning" Experiment across different data splits and models (macro-averaged AUROC)

model's performance on the validation set is much larger than on the test set (see Table 4.1), indicating the performance drop when testing on an unseen domain.

4.2 Results of the "Effect of Batch Size on Generalization" Experiment

	25%	50%	75%	100%
SE-ResNet DANN	0.90 (batch size = 128)	0.90 (batch size = 256)	0.88 (batch size = 256)	0.91 (batch size = 256)
SE-ResNet Baseline	0.91	0.89	0.91	0.90

Table 4.4: Testing results of the "Effect of Batch Size on Generalization" Experiment across different data splits (macro-averaged AUROC)

Table 4.4 shows the resulting values of the macro-averaged AUROC of the "Effect of Batch Size on Generalization" Experiment, where the SE-ResNet DANN model was tested on the CPSC and CPSC-Extra domain and compared to the SE-ResNet Baseline after selecting the optimal value of the batch size for each data split during validation. According to the table, 256 is the optimal batch size for the SE-ResNet DANN model trained on 50%, 75%, and 100% of the data, whereas 128 is optimal with 25% of the data. Also, from the table, it can be seen that the SE-ResNet DANN model outperformed the SE-ResNet Baseline model in the case of 50% and

100% of the data by 0.01, while the opposite happened with 25% and 75% of the data (by 0.01 and 0.03, respectively). Overall, the performance of the SE-ResNet DANN model is worse after doing a batch size selection with the validation set, and the model trained with the original batch size of 64 yields a better performance and outperforms the baseline model in most cases (see Section 4.1).

Table 4.5 shows the values of the macro-averaged AUROC of the "Effect of Batch Size on Generalization" Experiment, where several SE-ResNet DANN models, trained with different batch sizes, were tested on the CPSC and CPSC-Extra domain across different data splits. This is shown for illustrative purposes only. According to the table, there does not seem to be any relationship between the batch size and the test performance of the SE-ResNet DANN model on the unseen domain, since the performance values are different for each data split.

Batch Size	25%	50%	75%	100%
16	0.90	0.90	0.90	0.91
32	0.91	0.90	0.90	0.89
64	0.92	0.90	0.91	0.92
128	0.90	0.89	0.91	0.91
256	0.89	0.90	0.88	0.91

Table 4.5: Testing results of SE-ResNet DANN models trained with different batch sizes across different data splits (macro-averaged AUROC). This is shown for illustrative purposes only

4.3 Results of the "Effect of Learning Rate on Generalization" Experiment

Table 4.6 shows the resulting values of the macro-averaged AUROC of the "Effect of Learning Rate on Generalization" Experiment, where the SE-ResNet DANN model was tested on the CPSC and CPSC-Extra domain and compared to the SE-ResNet Baseline after selecting the optimal value of the learning rate for each data split

	25%	50%	75%	100%
SE-ResNet DANN	0.92 (learning rate = 0.003)	0.89 (learning rate = 0.001)	0.91 (learning rate = 0.001)	0.91 (learning rate = 0.001)
SE-ResNet Baseline	0.91	0.89	0.91	0.90

Table 4.6: Testing results of the "Effect of Learning Rate on Generalization" Experiment across different data splits (macro-averaged AUROC)

during validation. According to the table, 0.001 is the optimal learning rate for the SE-ResNet DANN model trained on 50%, 75%, and 100% of the data, whereas 0.003 is optimal with 25% of the data. Also, from the table, it can be seen that the SE-ResNet DANN model outperformed the SE-ResNet Baseline model only in the cases of 25% and 100% by 0.01, while, with the other data sizes, the performances were the same for both models. Overall, the performance of the SE-ResNet DANN model is worse after doing a learning rate selection with the validation set, and the model trained with the original learning rate of 0.003 yields a better performance and outperforms the baseline model in most cases (see Section 4.1).

Table 4.7 shows the values of the macro-averaged AUROC of the "Effect of Learning Rate on Generalization" Experiment, where several SE-ResNet DANN models, trained with different learning rates, were tested on the CPSC and CPSC-Extra domain across different data splits. This is shown for illustrative purposes only. According to the table, the test performance on the unseen domain peaks at the learning rate of around 0.001-0.003 and then drops at 0.1 for all the data splits. When the values are 0.00001, 0.0001, 0.01, and 0.1, the performance is worse than when the values are 0.001 and 0.003.

Learning Rate	25%	50%	75%	100%
0.00001	0.82	0.89	0.90	0.91
0.0001	0.83	0.89	0.88	0.88
0.001	0.90	0.89	0.91	0.91
0.003	0.92	0.90	0.91	0.92
0.01	0.90	0.87	0.88	0.90
0.1	0.73	0.75	0.77	0.72

Table 4.7: Testing results of SE-ResNet DANN models trained with different learning rates across different data splits (macro-averaged AUROC). This is shown for illustrative purposes only

Extreme value of λ	25%	50%	75%	100%
0.001	0.91	0.88	0.90	0.91
5	0.91	0.89	0.90	0.89
10	0.90	0.89	0.89	0.89
50	0.84	0.89	0.86	0.88
100	0.61	0.87	0.79	0.87
500	0.81	0.72	0.73	0.78
1000	0.89	0.69	0.67	0.60

Table 4.8: Testing results of SE-ResNet DANN models trained with various extreme values of λ across different data splits (macro-averaged AUROC). This is shown for illustrative purposes only

4.4 Results of the "Testing Extreme λ Values" Experiment

Table 4.8 shows the values of the macro-averaged AUROC of the "Testing Extreme λ Values" Experiment, where several SE-ResNet DANN models, trained with different extreme values of λ, were tested on the CPSC and CPSC-Extra domain across different data splits. This is shown for illustrative purposes only. According to the table, the test performance on the unseen domain is quite high and stable for the values of λ in the range of 0.001-10 across all data splits. Starting from the extreme value of 50, the performance starts to decline and becomes the worst at the value of 1000.

4.5 Results of the "Impact of Test Domain Selection" Experiment

This section shows the results of the "Impact of Test Domain Selection" Experiment, where different domains, other than CPSC and CPSC-Extra, were used as test sets. The SE-ResNet DANN and SE-ResNet Baseline were tested and compared in terms of their test performance on the unseen domain.

	25%	50%	75%	100%
SE-ResNet DANN	0.91	0.91	0.92	0.92
SE-ResNet Baseline	0.91	0.91	0.92	0.92

Table 4.9: Testing results of the "Impact of Test Domain Selection" Experiment across different data splits (macro-averaged AUROC), where the G12EC domain is used as a test set

Table 4.9 shows the resulting values of the macro-averaged AUROC when the SE-ResNet DANN and SE-ResNet Baseline models were tested on the G12EC domain and trained on all the other domains across all data splits. According to the table, the SE-ResNet DANN model did not outperform the SE-ResNet Baseline model across all data splits. With all the data splits, the performances of both models are the same.

	25%	50%	75%	100%
SE-ResNet DANN	0.95	0.96	0.97	0.96
SE-ResNet Baseline	0.95	0.95	0.95	0.95

Table 4.10: Testing results of the "Impact of Test Domain Selection" Experiment across different data splits (macro-averaged AUROC), where the SPH domain is used as a test set

Table 4.10 shows the resulting values of the macro-averaged AUROC when the SE-ResNet DANN and SE-ResNet Baseline models were tested on the SPH domain

and trained on all the other domains across all data splits. According to the table, the SE-ResNet DANN model outperformed the SE-ResNet Baseline model with 50% (by 0.01), 75% (by 0.02), and 100% (by 0.01) of the data. With 25% of the data, the performances of both models are the same.

	25%	50%	75%	100%
SE-ResNet DANN	0.92	0.93	0.93	0.93
SE-ResNet Baseline	0.92	0.93	0.93	0.93

Table 4.11: Testing results of the "Impact of Test Domain Selection" Experiment across different data splits (macro-averaged AUROC), where the PTB and PTB-XL domain is used as a test set

Table 4.11 shows the resulting values of the macro-averaged AUROC when the SE-ResNet DANN and SE-ResNet Baseline models were tested on the PTB and PTB-XL domain and trained on all the other domains across all data splits. According to the table, the values of the macro-averaged AUROC of both models are the same across all data splits. The SE-ResNet DANN model did not show any difference with the SE-ResNet Baseline model in terms of performance.

	25%	50%	75%	100%
SE-ResNet DANN	0.93	0.94	0.94	0.94
SE-ResNet Baseline	0.93	0.94	0.94	0.94

Table 4.12: Testing results of the "Impact of Test Domain Selection" Experiment across different data splits (macro-averaged AUROC), where the Chapman-Shaoxing and Ningbo domain is used as a test set

Table 4.12 shows the resulting values of the macro-averaged AUROC when the SE-ResNet DANN and SE-ResNet Baseline models were tested on the Chapman-Shaoxing and Ningbo domain and trained on all the other domains across all data splits. According to the table, the SE-ResNet DANN model did not outperform the SE-ResNet Baseline model across all data splits. With all data splits, the perfor-

mances of both models are the same.

Overall, comparing the results of this experiment, both the SE-ResNet DANN and SE-ResNet Baseline models yielded the worst test performance on the unseen domain when the CPSC and CPSC-Extra domain was used as a test set. The test performance is the second worst when the G12EC domain is used as a test set and the third worst when the PTB and PTB-XL domain is used. When the Chapman-Shaoxing and Ningbo domain is used as a test set, the test performance on the unseen domain is the second best. Finally, both models yield the best test performance on the unseen domain when the SPH domain is used as a test set. However, the SE-ResNet DANN model shows a performance improvement over the SE-ResNet Baseline model only when the SPH domain and the CPSC and CPSC-Extra domains are used as test sets.

4.6 Results of the "Tuning DANN for SPH Test Domain" Experiment

Table 4.13 shows the resulting values of the macro-averaged AUROC of the "Tuning DANN for SPH Test Domain" Experiment, where both the SE-ResNet DANN and SE-ResNet Baseline models were tested on the SPH domain across different data splits (25%, 50%, 75%, 100%). From the table, it can be seen that the SE-ResNet Baseline model outperformed the SE-ResNet DANN model with 25% and 100% of the data by 0.01. Meanwhile, with 50% and 75% of the data, the performances of both models are the same. Overall, the macro-averaged AUROC values of the SE-ResNet DANN model after the selection of the optimal value for λ are worse than those in the "Impact of Test Domain Selection" Experiment, where λ was not tuned, and testing was done directly.

Apart from the metric values, the table also shows the resulting values of the

	25%	50%	75%	100%
SE-ResNet DANN	0.94 ($\lambda = 0.01$)	0.95 ($\lambda = 0.01$)	0.95 ($\lambda = 1$)	0.94 ($\lambda = 0.01$)
SE-ResNet Baseline	0.95	0.95	0.95	0.95

Table 4.13: Testing results of the "Tuning DANN for SPH Test Domain" Experiment across different data splits (macro-averaged AUROC), using the SPH domain as a test set

hyperparameter λ of the SE-ResNet DANN model chosen during validation for each data split. According to the table, the selected value is the same for 25%, 50%, and 100% of the data ($\lambda = 0.01$), but different for 75% ($\lambda = 1$).

4.7 Results of the "Generalization Under Data Scarcity" Experiment

Table 4.14 shows the resulting values of the macro-averaged AUROC of the "Generalization Under Data Scarcity" Experiment, where both the SE-ResNet DANN and SE-ResNet Baseline models were tested on the SPH domain after having been trained on 5% and 10% of the data. Additionally, the table shows the results for the other splits (25%, 50%, 75%, 100%) for comparison. From the table, it can be seen that, with 10% of the data, the SE-ResNet DANN model outperformed the SE-ResNet Baseline model by 0.02. However, with 5% of the data, the SE-ResNet Baseline model outperformed the SE-ResNet DANN model by 0.01. Overall, according to the table, the performances of both models are worse with small training sets (5%, 10%) than with large ones (25%, 50%, 75%, 100%). The same is true when the other domains are used as test sets (see Table A.10 (the G12EC domain as a test set), Table A.11 (the CPSC and CPSC-Extra domain as a test set), Table A.12 (the PTB and PTB-XL domain as a test set), and Table A.13 (the Chapman-Shaoxing and Ningbo domain as a test set) in Appendix A). Moreover, in general, the SE-

ResNet DANN model does not show an improved performance over the SE-ResNet Baseline model with these extremely small training sets, since its macro-averaged AUROC values are mostly equal to or less than those of the SE-ResNet Baseline model.

	5%	10%	25%	50%	75%	100%
SE-ResNet DANN	0.92	0.94	0.95	0.96	0.97	0.96
SE-ResNet Baseline	0.93	0.92	0.95	0.95	0.95	0.95

Table 4.14: Testing results of the "Generalization Under Data Scarcity" Experiment across different data splits (macro-averaged AUROC), including 5% and 10%, where the SPH domain is used as a test set

4.8 Results of the "MixStyle vs. DANN and Baseline Comparison" Experiment

Table 4.15 shows the resulting values of the macro-averaged AUROC of the "MixStyle vs. DANN and Baseline Comparison" Experiment, where, using the value of 0.1 for the hyperparameter α, the SE-ResNet MixStyle model was tested on the SPH domain across different data splits (25%, 50%, 75%, 100%) and compared to the SE-ResNet DANN and SE-ResNet Baseline models. According to the table, the SE-ResNet MixStyle model did not outperform either model with all the data splits, yielding a worse performance than both models with 50%, 75%, and 100% of the data.

To try to improve the performance of the SE-ResNet MixStyle model, some hyperparameter selection was performed to select the optimal value of α for each data split. Table 4.16 shows the resulting values of the macro-averaged AUROC after performing the hyperparameter selection. From the table, it can be seen that the hyperparameter selection resulted in some improvements in the model's perfor-

	25%	50%	75%	100%
SE-ResNet MixStyle	0.95	0.92	0.93	0.94
SE-ResNet DANN	0.95	0.96	0.97	0.96
SE-ResNet Baseline	0.95	0.95	0.95	0.95

Table 4.15: Testing results of the "MixStyle vs. DANN and Baseline Comparison" Experiment across different data splits (macro-averaged AUROC), using $\alpha = 0.01$ for training and the SPH domain as a test set

	25%	50%	75%	100%
SE-ResNet MixStyle	0.95 ($\alpha = 0.1$)	0.94 ($\alpha = 0.3$)	0.94 ($\alpha = 0.2$)	0.92 ($\alpha = 0.3$)
SE-ResNet DANN	0.95	0.96	0.97	0.96
SE-ResNet Baseline	0.95	0.95	0.95	0.95

Table 4.16: Testing results of the "MixStyle vs. DANN and Baseline Comparison" Experiment across different data splits (macro-averaged AUROC), using the SPH domain as a test set, after selecting the optimal value of α for each data split

mance, but not consistently. In the end, the performance of the SE-ResNet MixStyle model is still worse than that of the two other models for all data splits.

5 Discussion

This chapter presents a thorough discussion of the results. First, there is a brief summary of the results from Chapter 4. Then, the results are interpreted in the context of all research questions. Subsequently, the results are compared to those from previous studies. Following that, there is a comprehensive critical analysis of the results and any interesting findings. The chapter also provides some limitations of the study. Finally, there is a discussion of the implications of the results for AI in healthcare and of future work that could be done to offer potential enhancements for better domain generalizability of deep learning models in ECG classification.

5.1 Summary of Findings

Based on the conducted experiments, described in Section 3.6, the results from Chapter 4 can be summarized as follows:

- The SE-ResNet DANN model showed an improvement in performance over the SE-ResNet Baseline model with almost all data splits (25%, 50%, 75%, and 100%) when the CPSC and CPSC-Extra and the SPH domains were used as test sets. When the G12EC, the PTB and PTB-XL, and the Chapman-Shaoxing and Ningbo domains were used as test sets, there were no improvements from the SE-ResNet DANN model.

- The SE-ResNet MixStyle model did not show any improvements in perfor-

mance over the SE-ResNet DANN and SE-ResNet Baseline models. In some cases, its performance was even worse than that of the two other models.

- The SE-ResNet DANN model yielded a worse performance when hyperparameters were selected with the validation set. When the default values of batch size and learning rate and the recommended value of the hyperparameter λ were used, the model yielded its best performance. When it comes to the SE-ResNet MixStyle model, the selection of the optimal value of the hyperparameter α resulted in an inconsistent improvement of the model's performance, but it was still worse than that of the two other models.

- The performance of the SE-ResNet DANN model on the test set was sensitive to the choice of hyperparameters, especially λ. The performance of the model deteriorated at extremely high values.

- The performance of the SE-ResNet DANN model significantly deteriorated when only 5% and 10% of the training data were used. Also, the SE-ResNet DANN model did not show an improved performance over the SE-ResNet Baseline model with these scarce training sets.

- The performance of the SE-ResNet Baseline model on an unseen domain is significantly worse than on a seen domain, as confirmed by the "Baseline vs. DANN with λ Tuning" Experiment (see Section 4.1). This is the case even for models with domain generalization methods, such as multi-source domain-adversarial training, where, despite some improvements, the performance on an unseen domain is still worse than on a seen domain.

5.2 Interpretation of Results

The findings of this thesis have provided answers to all the research questions stated in Section 1.4 as follows:

Research Question 1: Are domain generalization methods from recent literature, such as multi-source domain-adversarial training (DANN) and MixStyle, promising for improving generalizability in ECG classification? This question has been answered in Section 2.2 of Chapter 2. In particular, this section showed that the two domain generalization methods investigated in this study - multi-source domain-adversarial training (DANN) and MixStyle - are indeed promising for improving domain generalizability in ECG classification, since several studies successfully utilized these two methods in a similar context. Even though not so many studies on using these methods in the context of ECG classification have been found, the results of the identified studies demonstrated some potential of the two domain generalization methods for improving domain generalizability. For instance, in their studies, Hasani et al. [38] and Shang et al. [39] demonstrated the potential of multi-source domain-adversarial training (DANN), while Yang et al. [47] did that for the MixStyle method. Moreover, there are many studies that explore the theoretical framework behind these methods and other applications of them, also showing promising results. Although in the studies, these two methods demonstrate only moderate improvement in domain generalizability of ECG classification models, they still show some promise, so they were used in this thesis.

Research Question 2: To what extent do these domain generalization techniques improve the performance of deep learning models when evaluated on data from previously unseen sources? This question has been answered in the **"Baseline vs. DANN with λ Tuning"**, **"Impact of Test Domain Selection"**, **"Tuning DANN for SPH Test Domain"**, and **"MixStyle vs. DANN and Baseline Comparison"** Experiments (see Sections 4.1, 4.5, 4.6,

and 4.8). The results of these experiments demonstrate that multi-source domain-adversarial training (DANN) can be effective in improving domain generalization in the context of ECG classification, but this is the case only when the CPSC and CPSC-Extra and the SPH domains are used as test sets. In this case, the values of the macro-averaged AUROC showed improvements by 0.02 at most. With some training data portions, it was by 0.01, and, in rare cases, there was no improvement at all. When the G12EC, the PTB and PTB-XL, and the Chapman-Shaoxing and Ningbo domains were used as test sets, there were no visible improvements. This variability could be caused by domain similarity or variations in data distribution, which were not explicitly addressed in this study. Overall, multi-source domain-adversarial training (DANN) showed some promise in improving domain generalizability, but the effect is modest and context-dependent. The MixStyle method did not show any improvements in domain generalization in this study. These findings suggest that adversarial methods might be more promising than feature perturbation techniques like MixStyle in this context.

Research Question 3: How do factors such as the amount of training data and hyperparameter settings influence the effectiveness of these generalization methods? This question has been answered in the **"Baseline vs. DANN with λ Tuning"**, **"Effect of Batch Size on Generalization"**, **"Effect of Learning Rate on Generalization"**, **"Testing Extreme λ Values"**, **"Tuning DANN for SPH Test Domain"**, **"Generalization Under Data Scarcity"**, and **"MixStyle vs. DANN and Baseline Comparison"** Experiments (see Sections 4.1, 4.2, 4.3, 4.4, 4.6, 4.7, and 4.8). The results of these experiments confirmed that the weight hyperparameters (λ and α) and other hyperparameters (e.g., learning rate and batch size) affect the domain generalization performance. For instance, the ResNet DANN model showed some performance degradation with extremely high values of λ, highlighting the importance of careful hyperparameter tuning in

domain-adversarial training settings, as discussed in Section 3.4.2. Moreover, the experiments showed that the ResNet DANN model performed better with the default hyperparameter settings than with those selected with the validation set. This suggests that hyperparameter selection makes the performance of the model and its domain generalization worse, and it is not applicable in this context. Finally, as expected, the SE-ResNet DANN model showed a poorer performance and domain generalization with small training sets (5% and 10%), demonstrating the importance of having enough training data for good model performance and domain generalization. In most cases, the performances of both the SE-ResNet DANN and SE-ResNet MixStyle were high enough starting with 25% of training data.

5.3 Comparison to Previous Work

The results of this thesis are consistent with those of previous work. Specifically, the results show that, when no domain generalization methods are used, the performance of ECG deep learning models, trained on multi-source data, drops when they are tested on data from an unseen domain. This is consistent with the findings of Leinonen et al. [16], Han et al. [35], and A. Ballas and C. Diou [37]. Thus, the thesis confirms that even with multi-source data, domain generalization still remains a challenge and requires some additional techniques.

Moreover, the results of the thesis demonstrated that multi-source domain-adversarial training (DANN) shows some improvement in performance over the baseline model, but it is only marginal. This is consistent with the results of the works of Hasani et al. [38] and Shang et al. [39]. They also showed that this domain generalization method improves the performance of ECG deep learning models in an unseen domain only to a small extent. However, in this thesis, the SE-ResNet DANN model showed improvements only when two out of five domains were used as test sets. In the study of Hasani et al. [38], they tested the model with DANN on four

different domains and averaged them out, but they did not show the performance results for individual domains. Perhaps some of the results were unsatisfactory, but they did not want to show it. In the study of Shang et al. [39], they used only one unseen domain as a test set, and the performance improvement on it was minimal.

Thus, it is reasonable to assume that these authors cherry-picked their results to show some promise in their findings, but the actual results were less satisfactory. Furthermore, this claim can be supported by the fact that these are the only publicly available studies on the use of multi-source domain-adversarial training (DANN) in ECG classification, so maybe this is the case because the results are unsatisfactory, and this method does not work well in practice. Nevertheless, just like the results of Hasani et al. [38] and Shang et al. [39], the results of this thesis showed that multi-source domain-adversarial training (DANN) still shows some promise in improving domain generalizability in the context of ECG classification, but it is only part of the deep learning generalizability improvement. There may be some additional challenges that need to be addressed to achieve robust generalization.

Unlike this thesis, the studies by Hasani et al. [38] and Shang et al. [39] did not focus on the selection of the values of hyperparameters, such as λ, batch size, and learning rate, using the validation set. The hyperparameter selection in this thesis led to a worse performance of the SE-ResNet DANN model in most cases. Therefore, it is fair to assume that this is the reason why Hasani et al. [38] and Shang et al. [39] did not perform hyperparameter selection in the first place. Perhaps it is not suitable for improving the domain generalizability of deep learning models in the context of ECG classification.

When it comes to the MixStyle method, the results of this thesis demonstrated that this domain generalization method does not work well for improving domain generalization with this data and type of task. The performance of the model with MixStyle ended up being worse than that of the baseline model. This is not

consistent with the results of Yang et al. [47] since they claimed that the model with MixStyle performed better than the baseline model on the hidden validation set. This discrepancy in the results might come from the fact that the authors of the study used slightly different evaluation settings since they had "repeated scoring on the hidden validation set and one-time scoring on the hidden test set" [47]. The performance results on the hidden test set were pretty low, but the authors did not provide the performance results on the hidden test set of the other two models, so it was impossible to compare testing results between the three models. Thus, it is fair to assume that these results were also cherry-picked. This may also be supported by the fact that the study by Yang et al. [47] is the only study found on the use of MixStyle for ECG classification. Nevertheless, the results of this thesis showed that the MixStyle method does not work well in improving domain generalization with the given model, data, and task.

5.4 Critical Analysis of Results

The results of the experiments demonstrated that the SE-ResNet DANN model outperformed the SE-ResNet Baseline model only when tested on the CPSC and CPSC-Extra and the SPH domains. This tendency could be explained by the underlying domain properties of these datasets and the unique status of these domains. For instance, the CPSC and CPSC-Extra data source is the smallest of the five domains in terms of total ECG recordings. Thus, its underrepresentation during the training phase may have decreased overfitting to its specific domain characteristics, making it more sensitive to the domain-invariant representations encouraged by DANN. Moreover, the feature extractor may have found it easier to align representations across domains due to the relatively uniform ECG duration and sampling properties of the CPSC and CPSC-Extra domain. When it comes to the SPH domain, it can be considered an outlier among all the domains because it does not originate

from the George B. Moody PhysioNet Challenge 2021 dataset, unlike the other four sources. Therefore, this difference might have led to clearer domain boundaries, which could improve domain alignment by making it possible for the domain classifier of the DANN framework to more effectively differentiate between source and target domains.

Apart from that, the variations in domain generalization between different domains might be explained by certain demographic differences between them. For example, the SPH, the CPSC and CPSC-Extra, and the Chapman-Shaoxing and Ningbo data sources come from China, while the G12EC and the PTB and PTB-XL data sources come from more Western countries. Thus, they might differ from each other in terms of patient populations, clinical settings, ECG devices, and labeling practices. These differences might have impacted how well the models generalized across different domains.

The experiments demonstrated that the SE-ResNet MixStyle model did not show any improvements in performance and generalization over the SE-ResNet Baseline model. This might be explained by the fact that MixStyle incorporates stochastic perturbations in feature statistics (mean and variance) in order to help models become less dependent on surface-level domain-specific signals. However, in this ECG classification task, these signals may overlap with clinically significant diagnostic patterns. Thus, the use of the MixStyle method might have limited its efficiency by unintentionally suppressing important clinical features and domain-specific noise.

The results of the "Testing Extreme λ Values" Experiment (see Section 4.4), where the SE-ResNet DANN model was experimented with extreme values of the hyperparameter λ, demonstrated that the performance of the model deteriorated when extremely high values were used. This is consistent with the previous findings, discussed in Section 3.4.2, that the selection of the value of λ has an impact on how well the DANN loss function is optimized. Thus, it is important to carefully tune

hyperparameters in domain-adversarial training settings and find an appropriate balance between the classification loss and domain loss.

The experiments also demonstrated that hyperparameter selection using the validation set did not lead to improved generalization performance for both the SE-ResNet DANN and the SE-ResNet MixStyle models. In contrast, when default hyperparameter settings were used, both models yielded a better performance on the unseen domain. This might be explained by the fact that the validation set consists of data from domains that were already seen during training. Consequently, it might have encouraged overfitting to the common distributional characteristics of the training domains. Since the task is to generalize to new, unseen domains, it is more reasonable to use a separate unseen domain as a validation set, with hyperparameter selection based on leave-one-hospital-out cross-validation. In other words, in this thesis, the model's ability to generalize to new domains was not reliably shown by the validation results, and the performance on an unseen domain should be the basis for hyperparameter selection. However, in this thesis, there were only five domains, so using only three of them for training would probably not be enough to effectively produce robust, domain-invariant features. Nevertheless, it is worth experimenting with that in future research.

5.5 Limitations

There were several limitations in this thesis. First, only two domain generalization methods have been explored and experimented with here. There are some other methods worth exploring that could produce different results. Moreover, there were only five domains used in this study. By adding more domains, there would be the possibility of better hyperparameter selection through validation, and thus, potentially better testing results. Apart from that, only one deep learning architecture was used in this thesis. Therefore, there is a possibility that the results of this thesis

might not apply to other architectures. Finally, the differences between the domains used were not analyzed in detail. This analysis could provide explanations of why there were differences in performance and domain generalization between different domains.

5.6 Implications

This thesis has shown that current deep learning models for ECG classification, even if they are trained on multi-source data, still lack robustness when tested on data from unseen sources and hospitals. Furthermore, the results of the experiments showed that multi-source domain-adversarial training techniques (DANN) might effectively work to improve the domain generalizability of deep learning models in the context of ECG classification, but they are context-dependent and should be further explored and experimented with. These methods are not sufficient on their own, and additional research is required to further improve the robustness and generalization of deep learning models in ECG classification. Therefore, these models are not yet ready to be deployed in real-world clinical settings and need to be further refined to ensure that they work well with diverse data, especially from new, unseen hospitals.

5.7 Future Work

Overall, based on the findings and limitations identified in this thesis, several actions could be taken as part of future work.

First, future research could investigate some alternative domain generalization methods, such as MLDG [102], IRM [103], VREx [104], Deep CORAL [105], etc. These techniques, compared to DANN and MixStyle, might offer additional insights because they are based on fundamentally different assumptions, such as learning

invariant predictors or aligning correlations.

Second, different domain generalization methods could be combined. This might potentially enhance robustness across different hospitals. For example, combining DANN with MixStyle could help balance the advantages of stochastic feature perturbation and adversarial alignment.

Apart from that, as part of future research, since it is more reasonable to use a separate unseen domain as a validation set, the data splitting strategy could be modified. In particular, leave-two-domains-out schemes, in which three domains are used for training, one for validation, and one for testing, could be investigated. This might enhance the reliability of hyperparameter selection for domain generalization and provide a more accurate estimate of out-of-domain generalization performance. Moreover, this type of partitioning would be more feasible and yield more accurate estimates of generalization performance if the dataset were expanded with more domains.

Furthermore, a detailed analysis of the differences between domains could be done. For instance, differences in patient populations, clinical settings, ECG devices, and labeling practices could be analyzed. This analysis might provide an explanation for why certain deep learning models generalize better to some hospitals than others. As a result, the robustness of models to these differences could be improved.

Finally, the methods implemented and evaluated in this study could be explored and tested in other clinical applications, including EEG classification, chest X-ray classification, etc. The investigation of the effectiveness of the methods in related medical fields could help generalize the findings of this study beyond ECG classification and contribute to progress in medical AI.

6 Conclusion

This thesis examined the challenge of improving the cross-hospital generalizability of deep learning models for electrocardiogram (ECG) classification. Despite the fact that deep learning models have demonstrated excellent performance in within-hospital settings, domain shift makes it difficult for them to generalize to new, unseen hospitals.

In order to tackle this problem, the study investigated two domain generalization methods: multi-source domain-adversarial training (DANN) and MixStyle. These methods were implemented into a SE-ResNet model architecture, and their efficiency was evaluated in comparison to a baseline SE-ResNet model using a number of different experiments, which were designed to assess model performance on unseen domains. The models were trained, validated, and tested on publicly available ECG data from five different data sources, representing various hospitals.

The findings demonstrated that, in some situations, the SE-ResNet DANN model performed better than the baseline model, particularly when the CPSC and CPSC-Extra and the SPH domains were used as test sets. However, these improvements were modest and context-dependent. When it comes to the SE-ResNet MixStyle model, it showed no improvement in generalization and performed worse than the other two models. Moreover, hyperparameter selection using the validation set did not help in improving performance and generalization, probably because validation sets did not include any unseen domains.

In conclusion, even though multi-source domain-adversarial training techniques (DANN) show some promise in improving generalization to new hospitals, they are not sufficient on their own. The results of this thesis illustrate the complexity of the cross-hospital generalization problem in ECG classification and emphasize the necessity of conducting further research on this topic.

7 Declaration of AI Usage in the Thesis

I made use of AI while writing this thesis in order to improve its quality and help myself in coming up with relevant points and ideas. First, I used an AI-based typing assistant called Grammarly (https://app.grammarly.com/) which helped me with language revision by reviewing and correcting various spelling and grammar mistakes in the essay. Second, I used a chatbot based on a large language model, called ChatGPT (https://chat.openai.com/), to help myself in the writing process by writing sample conclusion and abstract. It gave me ideas of what I should write in these sections, and I wrote them using my own words and treated the conclusion and abstract by ChatGPT as examples. Moreover, I used ChatGPT to help myself come up with a plan for the thesis and any concepts and sections that I should write. Also, I used it for brainstorming to help myself in coming up with interesting and relevant points and ideas for the thesis. Apart from that, I used ChatGPT for pondering about the topics of domain generalization and AI in ECG classification. Finally, I used a paraphrasing tool called QuillBot (https://quillbot.com/) a little bit in order to help myself paraphrase some sentences from references. Sometimes I struggle with paraphrasing, so this is why I utilized QuillBot. Of course, I did not copy the paraphrased sentences directly from QuillBot but used them as examples of how I should paraphrase.

References

[1] ECG Waves. "The ecg leads: Electrodes, limb leads, chest (precordial) leads, and the 12-lead ecg". Accessed: 2025-01-20. (2025), [Online]. Available: https://ecgwaves.com/ekg-ecg-leads-electrodes-systems-limb-chest-precordial/ (visited on 01/20/2025).

[2] Online Biology Notes. "Electrocardiogram (ecg): Working principle, normal ecg wave, application of ecg". Accessed: 2025-01-20. (2025), [Online]. Available: https://www.onlinebiologynotes.com/electrocardiogram-ecg-working-principle-normal-ecg-wave-application-of-ecg/ (visited on 01/20/2025).

[3] Physio-Pedia Contributors. "Electrocardiogram". Accessed: 2025-01-20. (2025), [Online]. Available: https://www.physio-pedia.com/Electrocardiogram (visited on 01/20/2025).

[4] T. Reichlin, R. Abächerli, R. Twerenbold, *et al.*, "Advanced ecg in 2016: Is there more than just a tracing?", *Swiss Medical Weekly*, vol. 146, w14303, 2016, [CrossRef] [PubMed]. DOI: 10.4414/smw.2016.14303.

[5] B. Lüderitz and A. B. de Luna, "The history of electrocardiography", *Journal of Electrocardiology*, vol. 50, p. 539, 2017, [CrossRef] [PubMed]. DOI: 10.1016/j.jelectrocard.2017.07.014.

[6] N. Rafie, A. H. Kashou, and P. A. Noseworthy, "Ecg interpretation: Clinical relevance, challenges, and advances", *Hearts*, vol. 2, no. 4, pp. 505–513, 2021. DOI: 10.3390/hearts2040039.

[7] J. A. Drezner, S. Sharma, A. Baggish, M. Papadakis, M. G. Wilson, J. M. Prutkin, *et al.*, "International criteria for electrocardiographic interpretation in athletes: Consensus statement", *British Journal of Sports Medicine*, vol. 51, no. 9, pp. 704–731, 2017. DOI: 10.1136/bjsports-2016-097331.

[8] S. Ranka, M. Reddy, and A. Noheria, "Artificial intelligence in cardiovascular medicine", *Current Opinion in Cardiology*, vol. 36, pp. 26–35, 2021. DOI: 10.1097/HCO.0000000000000812.

[9] B. Ose, Z. Sattar, A. Gupta, C. Toquica, C. Harvey, and A. Noheria, "Artificial intelligence interpretation of the electrocardiogram: A state-of-the-art review", *Current Cardiology Reports*, vol. 26, no. 6, pp. 561–580, Jun. 2024, Epub 2024 May 16. DOI: 10.1007/s11886-024-02062-1.

[10] Y. Ansari, O. Mourad, K. Qaraqe, and E. Serpedin, "Deep learning for ecg arrhythmia detection and classification: An overview of progress for period 2017–2023", *Frontiers in Physiology*, vol. 14, p. 1 246 746, Sep. 2023. DOI: 10.3389/fphys.2023.1246746.

[11] G. Christopoulos, J. Graff-Radford, C. L. Lopez, *et al.*, "Artificial intelligence-electrocardiography to predict incident atrial fibrillation: A population-based study", *Circulation: Arrhythmia and Electrophysiology*, vol. 13, no. 12, e009355, Dec. 2020. DOI: 10.1161/CIRCEP.120.009355.

[12] V. Chandrasekar, M. Y. Ansari, A. V. Singh, S. Uddin, K. S. Prabhu, S. Dash, *et al.*, "Investigating the use of machine learning models to understand the drugs permeability across placenta", *IEEE Access*, vol. 11, pp. 52 726–52 739, 2023. DOI: 10.1109/ACCESS.2023.3272987.

[13] M. Chu, P. Wu, G. Li, W. Yang, J. L. Gutiérrez-Chico, and S. Tu, "Advances in diagnosis, therapy, and prognosis of coronary artery disease powered by deep learning algorithms", *JACC Asia*, vol. 3, pp. 1–14, 2023. DOI: 10.1016/j.jacasi.2022.12.005.

[14] Z. I. Attia, S. Kapa, F. Lopez-Jimenez, *et al.*, "Screening for cardiac contractile dysfunction using an artificial intelligence-enabled electrocardiogram", *Nature Medicine*, vol. 25, no. 1, pp. 70–74, Jan. 2019. DOI: 10.1038/s41591-018-0240-2.

[15] M. A. Khan and Y. Kim, "Cardiac arrhythmia disease classification using lstm deep learning approach", *Computers, Materials & Continua*, vol. 67, pp. 427–443, 2021. DOI: 10.32604/cmc.2021.014682.

[16] T. Leinonen, D. Wong, A. Vasankari, *et al.*, "Empirical investigation of multi-source cross-validation in clinical ecg classification", *Computers in Biology and Medicine*, vol. 183, p. 109 271, Dec. 2024, ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2024.109271.

[17] L. Alzubaidi, J. Bai, A. Al-Sabaawi, *et al.*, "A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications", *Journal of Big Data*, vol. 10, p. 46, 2023. DOI: 10.1186/s40537-023-00727-2.

[18] J. Yang, A. A. S. Soltan, and D. A. Clifton, "Machine learning generalizability across healthcare settings: Insights from multi-site covid-19 screening", *npj Digital Medicine*, vol. 5, p. 69, 2022. DOI: 10.1038/s41746-022-00614-9.

[19] E. Merdjanovska and A. Rashkovska, "Cross-database generalization of deep learning models for arrhythmia classification", in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, 2021, pp. 346–351. DOI: 10.23919/MIPRO52101.2021.9596930.

[20] M. F. Arslan, W. Guo, and S. Li, " Single-source Domain Generalization in Deep Learning Segmentation via Lipschitz Regularization ", in *proceedings of Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, vol. LNCS 15010, Springer Nature Switzerland, Oct. 2024.

[21] J. Krois, A. Garcia Cantu, A. Chaurasia, *et al.*, "Generalizability of deep learning models for dental image analysis", *Scientific Reports*, vol. 11, no. 1, p. 6102, Mar. 2021. DOI: 10.1038/s41598-021-85454-5.

[22] A. K. Ghosh, M. A. Unruh, S. Ibrahim, and M. F. Shapiro, "Association between patient diversity in hospitals and racial/ethnic differences in patient length of stay", *Journal of General Internal Medicine*, vol. 37, no. 4, pp. 723–729, Mar. 2022, Epub 2022 Jan 3. DOI: 10.1007/s11606-021-07239-w.

[23] M. Roschewitz, G. Khara, J. Yearsley, *et al.*, "Automatic correction of performance drift under acquisition shift in medical image classification", *Nature Communications*, vol. 14, p. 6608, 2023. DOI: 10.1038/s41467-023-42396-y.

[24] Z. Zhao, H. Fang, S. D. Relton, *et al.*, "Adaptive lead weighted resnet trained with different duration signals for classifying 12-lead ecgs", in *2020 Computing in Cardiology*, 2020, pp. 1–4. DOI: 10.22489/CinC.2020.112.

[25] Y. Ganin, E. Ustinova, H. Ajakan, *et al.*, *Domain-adversarial training of neural networks*, 2016. arXiv: 1505.07818 [stat.ML].

[26] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, *Domain generalization with mixstyle*, 2021. arXiv: 2104.02008 [cs.CV].

[27] 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*, vol. 25, pp. 65–69, 2019. DOI: 10.1038/s41591-018-0268-3.

[28] C. J. Van Rijsbergen, "A theoretical foundation for recall and precision", *Journal of Documentation*, vol. 30, no. 1, pp. 11–21, 1974.

[29] C. Zhang, G. Wang, J. Zhao, P. Gao, J. Lin, and H. Yang, "Patient-specific ecg classification based on recurrent neural networks and clustering technique", in *2017 13th IASTED International Conference on Biomedical Engineering (BioMed)*, 2017, pp. 63–67. DOI: 10.2316/P.2017.852-029.

[30] A. L. Goldberger, L. A. N. Amaral, L. Glass, *et al.*, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals", *Circulation*, vol. 101, no. 23, e215–e220, Jun. 2000. DOI: 10.1161/01.CIR.101.23.e215.

[31] Ö. Yildirim, "A novel wavelet sequence based on deep bidirectional lstm network model for ecg signal classification", *Computers in Biology and Medicine*, vol. 96, pp. 189–202, 2018, ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2018.03.016.

[32] Y. Xia, N. Wulan, K. Wang, and H. Zhang, "Atrial fibrillation detection using stationary wavelet transform and deep learning", in *2017 Computing in Cardiology (CinC)*, 2017, pp. 1–4. DOI: 10.22489/CinC.2017.210-084.

[33] A. Avetisyan, S. Tigranyan, A. Asatryan, *et al.*, *Deep neural networks generalization and fine-tuning for 12-lead ecg classification*, 2023. arXiv: 2305.18592 [eess.SP].

[34] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks", *Information Processing Management*, vol. 45, no. 4, pp. 427–437, 2009, ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2009.03.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457309000259.

[35] H. Han, S. Park, S. Min, *et al.*, "Towards high generalization performance on electrocardiogram classification", in *2021 Computing in Cardiology (CinC)*, vol. 48, 2021, pp. 1–4. DOI: 10.23919/CinC53138.2021.9662737.

[36] M. A. Reyna, N. Sadr, E. A. P. Alday, *et al.*, "Will two do? varying dimensions in electrocardiography: The physionet/computing in cardiology challenge 2021", in *2021 Computing in Cardiology (CinC)*, vol. 48, 2021, pp. 1–4. DOI: 10.23919/CinC53138.2021.9662687.

[37] A. Ballas and C. Diou, "A domain generalization approach for out-of-distribution 12-lead ecg classification with convolutional neural networks", in *2022 IEEE Eighth International Conference on Big Data Computing Service and Applications (BigDataService)*, 2022, pp. 9–13. DOI: 10.1109/BigDataService55688.2022.00009.

[38] H. Hasani, A. Bitarafan, and M. S. Baghshah, "Classification of 12-lead ecg signals with adversarial multi-source domain generalization", in *2020 Computing in Cardiology*, 2020, pp. 1–4. DOI: 10.22489/CinC.2020.445.

[39] Z. Shang, Z. Zhao, H. Fang, *et al.*, "Deep discriminative domain generalization with adversarial feature learning for classifying ecg signals", in *2021 Computing in Cardiology (CinC)*, vol. 48, 2021, pp. 1–4. DOI: 10.23919/CinC53138.2021.9662844.

[40] S. Lin, C.-T. Li, and A. C. Kot, "Multi-domain adversarial feature generalization for person re-identification", *IEEE Transactions on Image Processing*, vol. 30, pp. 1596–1607, 2021. DOI: 10.1109/TIP.2020.3046864.

[41] J. Wang, A. Wang, H. Hu, K. Wu, and D. He, "Multi-source domain generalization for ecg-based cognitive load estimation: Adversarial invariant and plausible uncertainty learning", in *ICASSP 2024 - 2024 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024, pp. 1631–1635. DOI: 10.1109/ICASSP48485.2024.10447676.

[42] L. Niu, C. Chen, H. Liu, S. Zhou, and M. Shu, "A deep-learning approach to ecg classification based on adversarial domain adaptation", *Healthcare (Basel)*, vol. 8, no. 4, p. 437, Oct. 2020. DOI: 10.3390/healthcare8040437.

[43] Y. Ganin and V. Lempitsky, *Unsupervised domain adaptation by backpropagation*, 2015. arXiv: 1409.7495 [stat.ML].

[44] M. HassanPour Zonoozi and V. Seydi, "A survey on adversarial domain adaptation", *Neural Processing Letters*, vol. 55, pp. 2429–2469, 2023. DOI: 10.1007/s11063-022-10977-5.

[45] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, *Domain-adversarial neural networks*, 2015. arXiv: 1412.4446 [stat.ML].

[46] H. Rangwani, S. K. Aithal, M. Mishra, A. Jain, and R. V. Babu, *A closer look at smoothness in domain adversarial training*, 2022. arXiv: 2206.08213 [cs.LG].

[47] H.-C. Yang, W.-T. Hsieh, and T. P.-C. Chen, "A mixed-domain self-attention network for multilabel cardiac irregularity classification using reduced-lead electrocardiogram", in *2021 Computing in Cardiology (CinC)*, vol. 48, 2021, pp. 01–04. DOI: 10.23919/CinC53138.2021.9662783.

[48] Y. Xiao, H. Yin, J. Bai, and R. K. Das, *Mixstyle based domain generalization for sound event detection with heterogeneous training data*, 2024. arXiv: 2407.03654 [eess.AS].

[49] E. A. Perez Alday, A. Gu, A. J. Shah, *et al.*, "Classification of 12-lead ecgs: The physionet/computing in cardiology challenge 2020", *Physiological Measurement*, Nov. 2020. DOI: 10.1088/1361-6579/abc960.

[50] F. Liu, C. Liu, L. Zhao, *et al.*, "An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection", *Journal of Medical Imaging and Health Informatics*, vol. 8, pp. 1368–1373, Sep. 2018. DOI: 10.1166/jmihi.2018.2442.

[51] P. Wagner, N. Strodthoff, R. D. Bousseljot, *et al.*, "Ptb-xl, a large publicly available electrocardiography dataset", *Scientific Data*, vol. 7, p. 154, 2020. DOI: 10.1038/s41597-020-0495-6.

[52] R. Bousseljot, D. Kreiseler, and A. Schnabel, "Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet", *Biomedical Engineering / Biomedizinische Technik*, vol. 40, no. s1, pp. 317–318, 1995. DOI: doi:10.1515/bmte.1995.40.s1.317.

[53] J. Zheng, H. Chu, D. Struppa, *et al.*, "Optimal multi-stage arrhythmia classification approach", *Scientific Reports*, vol. 10, p. 2898, 2020. DOI: 10.1038/s41598-020-59821-7.

[54] H. Liu, D. Chen, D. Chen, *et al.*, "A large-scale multi-label 12-lead electrocardiogram database with standardized diagnostic statements", *Scientific Data*, vol. 9, p. 272, 2022. DOI: 10.1038/s41597-022-01403-5.

[55] J. Zheng, J. Zhang, S. Danioko, *et al.*, "A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients", *Scientific Data*, vol. 7, p. 48, 2020. DOI: 10.1038/s41597-020-0386-x.

[56] V. Tihonenko, A. Khaustov, S. Ivanov, A. Rivin, and E. Yakushenko, *St petersburg incart 12-lead arrhythmia database*, 2008. [Online]. Available: https://physionet.org/content/incartdb/1.0.0/.

[57] Healio Cardiology, *First degree av block*, Accessed: February 4, 2025, n.d. [Online]. Available: https://www.healio.com/cardiology/learn-the-heart/cardiology-review/topic-reviews/first-degree-av-block.

[58] Centers for Disease Control and Prevention (CDC), *About other conditions related to heart disease*, Accessed: February 4, 2025, n.d. [Online]. Available: https://www.cdc.gov/heart-disease/about/other-conditions-related-to-heart-disease.html.

[59] M. Zoni-Berisso, F. Lercari, T. Carazza, and S. Domenicucci, "Epidemiology of atrial fibrillation: European perspective", *Clinical Epidemiology*, vol. 6, pp. 213–220, Jun. 2014. DOI: 10.2147/CLEP.S47385.

[60] Johns Hopkins Medicine, *Atrial flutter*, Accessed: February 4, 2025, n.d. [Online]. Available: https://www.hopkinsmedicine.org/health/conditions-and-diseases/atrial-flutter.

[61] M. S. Link, "Evaluation and initial treatment of supraventricular tachycardia", *New England Journal of Medicine*, vol. 367, no. 15, pp. 1438–1448, 2012. DOI: 10.1056/NEJMcp1111259.

[62] Cleveland Clinic, *Right bundle branch block*, Accessed: February 4, 2025, n.d. [Online]. Available: https://my.clevelandclinic.org/health/diseases/21692-right-bundle-branch-block.

[63] Cleveland Clinic, *Left anterior fascicular block*, Accessed: February 4, 2025, n.d. [Online]. Available: https://my.clevelandclinic.org/health/diseases/23212-left-anterior-fascicular-block.

[64] Sunfox, *Understanding left axis deviation ecg: Causes, diagnosis, and treatment*, Accessed: February 4, 2025, n.d. [Online]. Available: https://sunfox.in/blogs/left-axis-deviation-ecg/.

[65] A. H. Kashou, P. Shams, and L. Chhabra, *Electrical Right and Left Axis Deviation*, Updated 2024 Jan 8. Treasure Island (FL): StatPearls Publishing, 2024. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK470532/.

[66] Cleveland Clinic, *Left bundle branch block*, Accessed: February 4, 2025, n.d. [Online]. Available: https://my.clevelandclinic.org/health/diseases/23287-left-bundle-branch-block.

[67] Life in the Fast Lane, *Low qrs voltage*, Accessed: February 4, 2025, n.d. [Online]. Available: https://litfl.com/low-qrs-voltage-ecg-library/.

[68] Healio Cardiology, *Low voltage ecg review*, Accessed: February 4, 2025, n.d. [Online]. Available: https://www.healio.com/cardiology/learn-the-heart/ecg-review/ecg-topic-reviews-and-criteria/low-voltage-review.

[69] Cleveland Clinic, *Premature atrial contractions*, Accessed: February 4, 2025, n.d. [Online]. Available: https://my.clevelandclinic.org/health/diseases/21700-premature-atrial-contractions.

[70] Taylor and Francis, *Right axis deviation*, Accessed: February 4, 2025, n.d. [Online]. Available: https://taylorandfrancis.com/knowledge/Medicine_and_healthcare/Cardiology/Right_axis_deviation/.

[71] Wikipedia contributors, *Right axis deviation*, *Wikipedia, The Free Encyclopedia*, Accessed: February 4, 2025, n.d. [Online]. Available: https://en.wikipedia.org/wiki/Right_axis_deviation.

[72] J. R. Hampton, *The ECG Made Easy*, 8th. Edinburgh: Churchill Livingstone, 2013, p. 4, ISBN: 9780702046421.

[73] M. Gertsch, *The ECG: A Two-Step Approach to Diagnosis*, 1st. Springer-Verlag Berlin Heidelberg, 2004, pp. 19–21, ISBN: 978-3-540-00869-9. DOI: 10.1007/978-3-662-10315-9.

[74] UpToDate, *Normal sinus rhythm and sinus arrhythmia*, Accessed: February 4, 2025, n.d. [Online]. Available: https://www.uptodate.com/contents/normal-sinus-rhythm-and-sinus-arrhythmia.

[75] Cleveland Clinic, *Sinus arrhythmia*, Accessed: February 4, 2025, n.d. [Online]. Available: https://my.clevelandclinic.org/health/diseases/21666-sinus-arrhythmia.

[76] Cleveland Clinic, *Sinus bradycardia*, Accessed: February 4, 2025, n.d. [Online]. Available: https://my.clevelandclinic.org/health/diseases/22473-sinus-bradycardia.

[77] CardioSmart, *Bradycardia: Signs and symptoms*, Accessed: February 4, 2025, n.d. [Online]. Available: https://www.cardiosmart.org/topics/bradycardia/signs-and-symptoms.

[78] Cleveland Clinic, *Sinus tachycardia*, Accessed: February 4, 2025, n.d. [Online]. Available: https://my.clevelandclinic.org/health/diseases/23210-sinus-tachycardia.

[79] B. Saltin, *Exercise and Circulation in Health and Disease*, 1st. Human Kinetics, 2000, ch. 21, Chapter 21: Circulatory Regulation in Muscle Disease, ISBN: 978-0-88011-632-9.

[80] C. Haarmark, C. Graff, M. P. Andersen, *et al.*, "Reference values of electrocardiogram repolarization variables in a healthy population", *Journal of Electrocardiology*, vol. 43, no. 1, pp. 31–39, 2010, ISSN: 0022-0736. DOI: https://doi.org/10.1016/j.jelectrocard.2009.08.001.

[81] P. M. Rautaharju, B. Surawicz, and L. S. Gettes, "Aha/accf/hrs recommendations for the standardization and interpretation of the electrocardiogram: Part iv: The st segment, t and u waves, and the qt interval a scientific statement from the american heart association electrocardiography and arrhythmias committee, council on clinical cardiology; the american college of cardiology foundation; and the heart rhythm society endorsed by the international society for computerized electrocardiology", *Journal of the American*

College of Cardiology, vol. 53, no. 11, pp. 982–991, 2009, ISSN: 0735-1097. DOI: https://doi.org/10.1016/j.jacc.2008.12.014.

[82] M. D. Jacobsen, G. S. Wagner, L. Holmvang, *et al.*, "Clinical significance of abnormal t waves in patients with non-st-segment elevation acute coronary syndromes", *American Journal of Cardiology*, vol. 88, no. 11, pp. 1225–1229, Dec. 2001. DOI: 10.1016/s0002-9149(01)02081-1.

[83] F. I. Marcus and W. Zareba, "The electrocardiogram in right ventricular cardiomyopathy/dysplasia. how can the electrocardiogram assist in understanding the pathologic and functional changes of the heart in this disease?", *Journal of Electrocardiology*, vol. 42, no. 2, 136.e1–136.e5, 2009, ISSN: 0022-0736. DOI: https://doi.org/10.1016/j.jelectrocard.2008.12.011.

[84] R. Nash, "An introduction to convolutional neural networks", *arXiv*, 2015. arXiv: 1511.08458 [cs.NE].

[85] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016, pp. 326–366, ISBN: 978-0262035613.

[86] A. Zhang, Z. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning*. Cambridge, New York, Port Melbourne, New Delhi, Singapore: Cambridge University Press, 2024, Chapter 7.2: Convolutions for Images, ISBN: 978-1-009-38943-3.

[87] Towards Data Science, *Batch norm explained visually: How it works and why neural networks need it*, Accessed: February 4, 2025, n.d. [Online]. Available: https://towardsdatascience.com/batch-norm-explained-visually-how-it-works-and-why-neural-networks-need-it-b18919692739/.

[88] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *arXiv*, 2015. arXiv: 1502.03167 [cs.LG].

[89] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV].

[90] S. Basodi, C. Ji, H. Zhang, and Y. Pan, *Gradient amplification: An efficient way to train deep neural networks*, 2020. arXiv: 2006.10560 [cs.LG].

[91] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors*, 2012. arXiv: 1207.0580 [cs.NE].

[92] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks", in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141. DOI: 10.1109/CVPR.2018.00745.

[93] Aporia, *Understanding binary cross-entropy and log loss for effective model monitoring*, Accessed: March 18, 2025, 2023. [Online]. Available: https://www.aporia.com/learn/understanding-binary-cross-entropy-and-log-loss-for-effective-model-monitoring/.

[94] T. P. Lillicrap, A. Santoro, L. Marris, *et al.*, "Backpropagation and the brain", *Nature Reviews Neuroscience*, vol. 21, pp. 335–346, 2020. DOI: 10.1038/s41583-020-0277-3.

[95] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms", *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997, ISSN: 0031-3203. DOI: https://doi.org/10.1016/S0031-3203(96)00142-2.

[96] C. E. Metz, "Basic principles of roc analysis", *Seminars in Nuclear Medicine*, vol. 8, no. 4, pp. 283–298, 1978, ISSN: 0001-2998. DOI: https://doi.org/10.1016/S0001-2998(78)80014-2.

[97] T. Fawcett, "An introduction to roc analysis", *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, ROC Analysis in Pattern Recognition, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2005.10.010.

[98] P. Flach, J. Hernández-Orallo, and C. Ferri, "A coherent interpretation of auc as a measure of aggregated classification performance", in *Proceedings of the 28th International Conference on Machine Learning (ICML'11)*, Bellevue, Washington, USA: Omnipress, 2011, pp. 657–664, ISBN: 9781450306195.

[99] J. Huang and C. Ling, "Using auc and accuracy in evaluating learning algorithms", *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005. DOI: 10.1109/TKDE.2005.50.

[100] scikit-learn developers, *Multiclass receiver operating characteristic (roc)*, Accessed: February 4, 2025, n.d. [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html.

[101] G. Wu, C. Li, and Y. Yin, *Towards understanding generalization of macro-auc in multi-label learning*, 2023. arXiv: 2305.05248 [cs.LG].

[102] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, *Learning to generalize: Meta-learning for domain generalization*, 2017. arXiv: 1710.03463 [cs.LG].

[103] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, *Invariant risk minimization*, 2020. arXiv: 1907.02893 [stat.ML].

[104] D. Krueger, E. Caballero, J.-H. Jacobsen, *et al.*, *Out-of-distribution generalization via risk extrapolation (rex)*, 2021. arXiv: 2003.00688 [cs.LG].

[105] B. Sun and K. Saenko, *Deep coral: Correlation alignment for deep domain adaptation*, 2016. arXiv: 1607.01719 [cs.CV].

Appendix A Tables

Study	Database	# Cl	Classifier	Acc (%)	Se (%)	Sp (%)
Luo et al. (2017)	MIT-BIH	4	DNN-SDA	98.80	71.40	99.80
Majumdar and Ward (2017)	MIT-BIH	4	SVM-RBF	97.00	100.0	90.12
Zhang et al. (2017)	MIT-BIH	5	RNN	99.40	97.60	99.70
Xia et al. (2017)	MIT-BIH	3	CNN	98.63	98.79	97.87
Nguyen et al. (2018)	CUDB MIT-BIH (VFDB)	2	FCN	99.26	97.07	99.44
Jun et al. (2018)	MIT-BIH	4	2D CNN	99.05	99.57	97.85
Yildirim (2018)	MIT-BIH	4	Bi-directional LSTM	99.39	95.66	98.11
Sannino and De Pietro (2018)	MIT-BIH	4	DNN	99.68	99.48	99.83
Faust et al. (2018)	MIT-BIH	5	BiLSTM	98.51	98.32	98.67
Xia and Xie (2019)	MIT-BIH	4	1D CNN + Active Learning	99.20	95.73	98.73
Lui and Chow (2018)	MIT-BIH	4	ML-CNN	96.00	95.40	97.37
Xia et al. (2018)	MIT-BIH Wearable Device	4	DNN	99.80	99.40	99.90
Wang et al. (2019)	MIT-BIH	2	GRNN	97.40	86.70	98.30
Hanbay (2019)	MIT-BIH	4	DNN	96.40	86.41	96.41
Wang and Zhou (2019)	BIDMC-CHF + MIT-BIH NSR + Fantasia	5	LSTM	99.22	99.22	99.72
Chen et al. (2020)	MIT-BIH	4	CNN-LSTM	99.32	97.50	98.70
Fu et al. (2020)	PTB	6	CNN-BiGRUt	99.11	99.02	98.23
Sharma et al. (2021)	MIT-BIH	5	SVM + FFBPNN	98.53	98.24	95.68
Ojha et al. (2022)	MIT-BIH	4	CNN-SVM	99.53	98.24	97.58
Sepahvand and Abdali-Mohammadi (2022)	Chapman ECG DB	12	Distilled Models	98.15	97.11	98.45
Midani et al. (2023)	MIT-BIH	5	CNN + BiLSTM	99.46	97.01	99.57
Kumar et al. (2023)	MIT-BIH	5	Fuzz-ClustNet	98.66	98.92	93.88

Table A.1: An overview of deep learning models for the detection and classification of ECG arrhythmias, created by Ansari et al. [10]

Different Data Splits			
25% of Data	50% of Data	75% of Data	100% of Data
TRAINING:			
CPSC: 1146 **SPH:** 4376 **PTB:** 4001 **Chapman:** 8211	**CPSC:** 2291 **SPH:** 8752 **PTB:** 8003 **Chapman:** 16422	**CPSC:** 3437 **SPH:** 13128 **PTB:** 12004 **Chapman:** 24632	**CPSC:** 4582 **SPH:** 17504 **PTB:** 16005 **Chapman:** 32843
VALIDATION:			
CPSC: 382 **SPH:** 1443 **PTB:** 1336 **Chapman:** 2743	**CPSC:** 764 **SPH:** 2885 **PTB:** 2672 **Chapman:** 5486	**CPSC:** 1146 **SPH:** 4328 **PTB:** 4007 **Chapman:** 8228	**CPSC:** 1528 **SPH:** 5770 **PTB:** 5343 **Chapman:** 10971

Table A.2: Numbers of ECG recordings in each domain used for training and validation across different data splits, when the G12EC domain is used as a test set

Different Data Splits			
25% of Data	50% of Data	75% of Data	100% of Data
TRAINING:			
G12EC: 1668 **CPSC:** 1146 **PTB:** 4001 **Chapman:** 8211	**G12EC:** 3314 **CPSC:** 2291 **PTB:** 8003 **Chapman:** 16422	**G12EC:** 5004 **CPSC:** 3437 **PTB:** 12004 **Chapman:** 24632	**G12EC:** 6672 **CPSC:** 4582 **PTB:** 16005 **Chapman:** 32843
VALIDATION:			
G12EC: 555 **CPSC:** 382 **PTB:** 1336 **Chapman:** 2743	**G12EC:** 1110 **CPSC:** 764 **PTB:** 2672 **Chapman:** 5486	**G12EC:** 1665 **CPSC:** 1146 **PTB:** 4007 **Chapman:** 8228	**G12EC:** 2220 **CPSC:** 1528 **PTB:** 5343 **Chapman:** 10971

Table A.3: Numbers of ECG recordings in each domain used for training and validation across different data splits, when the SPH domain is used as a test set

Different Data Splits			
25% of Data	50% of Data	75% of Data	100% of Data
TRAINING:			
G12EC: 1668 **SPH:** 4376 **CPSC:** 1146 **Chapman:** 8211	**G12EC:** 3314 **SPH:** 8752 **CPSC:** 2291 **Chapman:** 16422	**G12EC:** 5004 **SPH:** 13128 **CPSC:** 3437 **Chapman:** 24632	**G12EC:** 6672 **SPH:** 17504 **CPSC:** 4582 **Chapman:** 32843
VALIDATION:			
G12EC: 555 **SPH:** 1443 **CPSC:** 382 **Chapman:** 2743	**G12EC:** 1110 **SPH:** 2885 **CPSC:** 764 **Chapman:** 5486	**G12EC:** 1665 **SPH:** 4328 **CPSC:** 1146 **Chapman:** 8228	**G12EC:** 2220 **SPH:** 5770 **CPSC:** 1528 **Chapman:** 10971

Table A.4: Numbers of ECG recordings in each domain used for training and validation across different data splits, when the PTB and PTB-XL domain is used as a test set

Different Data Splits			
25% of Data	50% of Data	75% of Data	100% of Data
TRAINING:			
G12EC: 1668 **SPH:** 4376 **PTB:** 4001 **CPSC:** 1146	**G12EC:** 3314 **SPH:** 8752 **PTB:** 8003 **CPSC:** 2291	**G12EC:** 5004 **SPH:** 13128 **PTB:** 12004 **CPSC:** 3437	**G12EC:** 6672 **SPH:** 17504 **PTB:** 16005 **CPSC:** 4582
VALIDATION:			
G12EC: 555 **SPH:** 1443 **PTB:** 1336 **CPSC:** 382	**G12EC:** 1110 **SPH:** 2885 **PTB:** 2672 **CPSC:** 764	**G12EC:** 1665 **SPH:** 4328 **PTB:** 4007 **CPSC:** 1146	**G12EC:** 2220 **SPH:** 5770 **PTB:** 5343 **CPSC:** 1528

Table A.5: Numbers of ECG recordings in each domain used for training and validation across different data splits, when the Chapman-Shaoxing and Ningbo domain is used as a test set

Different Data Splits	
5% of Data	10% of Data
TRAINING:	
CPSC: 229 **SPH:** 875 **PTB:** 800 **Chapman:** 1642	**CPSC:** 458 **SPH:** 1750 **PTB:** 1600 **Chapman:** 3284
VALIDATION:	
CPSC: 76 **SPH:** 289 **PTB:** 267 **Chapman:** 549	**CPSC:** 153 **SPH:** 577 **PTB:** 534 **Chapman:** 1097

Table A.6: Numbers of ECG recordings in each domain used for training and validation across 5% and 10% data splits, when the G12EC domain is used as a test set

Different Data Splits	
5% of Data	10% of Data
TRAINING:	
G12EC: 334 **SPH:** 875 **PTB:** 800 **Chapman:** 1642	**G12EC:** 667 **SPH:** 1750 **PTB:** 1600 **Chapman:** 3284
VALIDATION:	
G12EC: 111 **SPH:** 289 **PTB:** 267 **Chapman:** 549	**G12EC:** 222 **SPH:** 577 **PTB:** 534 **Chapman:** 1097

Table A.7: Numbers of ECG recordings in each domain used for training and validation across 5% and 10% data splits, when the CPSC and CPSC-Extra domain is used as a test set

Different Data Splits	
5% of Data	10% of Data
TRAINING:	
G12EC: 334 **SPH:** 875 **CPSC:** 229 **Chapman:** 1642	**G12EC:** 667 **SPH:** 1750 **CPSC:** 458 **Chapman:** 3284
VALIDATION:	
G12EC: 111 **SPH:** 289 **CPSC:** 76 **Chapman:** 549	**G12EC:** 222 **SPH:** 577 **CPSC:** 153 **Chapman:** 1097

Table A.8: Numbers of ECG recordings in each domain used for training and validation across 5% and 10% data splits, when the PTB and PTB-XL domain is used as a test set

Different Data Splits	
5% of Data	10% of Data
TRAINING:	
G12EC: 334 **SPH:** 875 **PTB:** 800 **CPSC:** 229	**G12EC:** 667 **SPH:** 1750 **PTB:** 1600 **CPSC:** 458
VALIDATION:	
G12EC: 111 **SPH:** 289 **PTB:** 267 **CPSC:** 76	**G12EC:** 222 **SPH:** 577 **PTB:** 534 **CPSC:** 153

Table A.9: Numbers of ECG recordings in each domain used for training and validation across 5% and 10% data splits, when the Chapman-Shaoxing and Ningbo domain is used as a test set

	5%	10%	25%	50%	75%	100%
SE-ResNet DANN	0.86	0.89	0.91	0.91	0.92	0.92
SE-ResNet Baseline	0.86	0.88	0.91	0.91	0.92	0.92

Table A.10: Testing results of the "Generalization Under Data Scarcity" Experiment across different data splits (macro-averaged AUROC), including 5% and 10%, where the G12EC domain is used as a test set

	5%	10%	25%	50%	75%	100%
SE-ResNet DANN	0.86	0.88	0.92	0.90	0.91	0.92
SE-ResNet Baseline	0.84	0.89	0.91	0.89	0.91	0.90

Table A.11: Testing results of the "Generalization Under Data Scarcity" Experiment across different data splits (macro-averaged AUROC), including 5% and 10%, where the CPSC and CPSC-Extra domain is used as a test set

	5%	10%	25%	50%	75%	100%
SE-ResNet DANN	0.88	0.89	0.92	0.93	0.93	0.93
SE-ResNet Baseline	0.88	0.91	0.92	0.93	0.93	0.93

Table A.12: Testing results of the "Generalization Under Data Scarcity" Experiment across different data splits (macro-averaged AUROC), including 5% and 10%, where the PTB and PTB-XL domain is used as a test set

	5%	10%	25%	50%	75%	100%
SE-ResNet DANN	0.84	0.88	0.93	0.94	0.94	0.94
SE-ResNet Baseline	0.84	0.89	0.93	0.94	0.94	0.94

Table A.13: Testing results of the "Generalization Under Data Scarcity" Experiment across different data splits (macro-averaged AUROC), including 5% and 10%, where the Chapman-Shaoxing and Ningbo domain is used as a test set