scientific reports

Check for updates

OPEN

Transformer-based heart language model with electrocardiogram annotations

Stojancho Tudjarski[1,2,4], Marjan Gusev[1,2,4]✉ & Evangelos Kanoulas[3,4]

This paper explores the potential of transformer-based foundation models to detect Atrial Fibrillation (AFIB) in electrocardiogram (ECG) processing, an arrhythmia specified as an irregular heart rhythm without patterns. We construct a language with tokens from heartbeat locations to detect irregular heart rhythms by applying a transformers-based neural network architecture previously used only for building natural language models. Our experiments include 41, 128, 256, and 512 tokens, representing parts of ECG recordings after tokenization. The method consists of training the foundation model with annotated benchmark databases, then finetuning on a much smaller dataset and evaluating different ECG datasets from those used in the finetuning. The best-performing model achieved an F1 score of 93.33 % to detect AFIB in an ECG segment composed of 41 heartbeats by evaluating different training and testing ECG benchmark datasets. The results showed that a foundation model trained on a large data corpus could be finetuned using a much smaller annotated dataset to detect and classify arrhythmia in ECGs. This work paves the way for the transformation of foundation models into invaluable cardiologist assistants soon, opening the possibility of training foundation models with even more data to achieve even better performance scores.

Cardiac disorders, especially atrial fibrillation (AFIB)[1], remain among the leading causes of morbidity and mortality worldwide. Early detection of AFIB episodes is essential for delivering a timely intervention and reducing possible complications, including stroke and heart failure. Technology based on electrocardiograms (ECGs) has been the precious standard for monitoring and diagnosing various cardiac abnormalities. Various sinus node, atrial, junctional, and ventricular arrhythmias present abnormal heart function and the Normal Sinus Rhythm (NSR) corresponds to normal heart activity. This paper aims to detect irregularities with the AFIB rhythm versus any other rhythm (non-AFIB). Detecting AFIB in ECGs is a task that finds if the heart rhythm is without any patterns, where the heart beats occasionally without any predefined regularity and sequence of electrical polarization and depolarization cycles. Introducing small wearable measuring devices creates an excellent opportunity for automated, accurate, and efficient arrhythmia detection and classification.

This problem was analyzed using many signal processing and machine learning (ML) techniques[2,3]. Still, no solution proved to be sufficiently precise, especially in the cases of new patients for which the models were not trained[4]. Our approach uses transformer-based foundation models[5] emerging as powerful tools in text and image processing, demonstrating unparalleled success in understanding complex patterns within large datasets. The main advantage against previously used ML techniques is capturing the context of a given input, considering the meaningful, relevant surrounding parts around any single data point (a single token representing a short ECG segment in our case).

Although GPT-2[6] and later versions as unidirectional transformers grab a lot of attention and provide a wide variety of benefits for the next most probable token in a given sequence, they are unsuited for classification tasks. To detect arrhythmia, we used the bidirectional RoBERTa model[7] as a variation of BERT (the first Transformers-based model for text processing[8]), widely accepted as the best performing among other variants.

Transfer learning[9] is the methodology of training neural network (NN) models to learn quickly from smaller datasets, starting from an already trained model with a massive amount of data, such as foundation models based on BERT/RoBERTa-based models. For example, these foundation models are finetuned for text classification tasks (sentiment analysis)[10–12]. During initial training in a self-supervised manner, foundation models capture complex data patterns, connections, and subtleties from a vast amount of unlabelled data, revealing a broad grasp of the inherited data structure. Further training of the already captured data patterns (finetuning) requires

[1]Innovation Dooel, 1000 Skopje, North Macedonia. [2]Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, 1000 Skopje, North Macedonia. [3]Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands. [4]Stojancho Tudjarski, Marjan Gusev and Evangelos Kanoulas contributed equally to this work. ✉email: marjan.gushev@finki.ukim.mk

a smaller volume of labeled data on fundamental features, adjusting the pre-learned knowledge to suit the specific details of the downstream task in a supervised manner.

The main objective of this study is to explore the potential of transformer-based foundation models for detecting and classifying arrhythmia using ECGs as input. In particular, we aim to find experimentally the optimal model for detecting AFIB as a proof-of-concept approach. Reaching the main objective validates the research hypothesis that *A bidirectional Transformers-based foundation model trained with an enormous annotated ECG dataset can be finetuned well enough with a much smaller labeled training dataset to detect AFIB rhythm*. Furthermore, we aim to determine whether the transformers' architecture, which has been proven highly performing in Natural Language Processing (NLP), can be applied in other areas with different data they operate in, such as ECG record processing.

We trained foundational models using information stored in a massive corpus of ECG recordings from 12 public benchmark ECG databases with more than 100 GB of data[13] and 22M heartbeats annotated manually by cardiology experts. Our task was to detect whether a given heartbeat sequence specifies an AFIB rhythm. The finetuning phase was trained on the MIT-BIH Arrhythmia ECG benchmark database (MITDB)[14] and used the MIT-BIH Atrial Fibrillation Database (AFDB)[15] and Long-Term AFIB Database (LTAFDB)[16] for performance evaluation. This method ensured that the model evaluation used a dataset different from the one used for training and finetuning. The results provide sufficient information to ensure relevant conclusions about the model's behavior compared to unobserved data during testing.

Our approach translates an ECG recording into text with an alphabet produced to allow the use of well-established tools for training large language models (LLMs), particularly foundation models. A *context window* refers to the maximum number of tokens that an LLM considers when generating text as a measure to understand the current input and develop an appropriate response. We conducted four experiments with 41, 128, 256, and 512 tokens passing to models in the first training phase in an unsupervised manner. The input passed to the models in the finetuning phase is a series of tokens representing 41 consecutive heartbeats, based on an enormous number of previously conducted experiments to detect the relevance of AFIB, which approximately corresponds to the 30-sec gold standard definition of AFIB as a clinically meaningful arrhythmia pattern[17,18]. Figure 1 illustrates AFIB and NSR rhythms. This research significantly contributes to the progress of foundation models within cardiac healthcare, advocating for their enhanced training and increased application.

Methods

This section describes the training and testing datasets, feature engineering, data transformations, development of the foundation model, and finetuning (Fig. 5), followed by the specifications of the experiments and evaluation approaches.

Datasets

Training of the foundation model requires a comprehensive set of ECG benchmark datasets that we collected by aggregating twelve public ECG databases: AFDB[15], AHADB[19], CUDB[20], EDB[21], INCARTDB[22], LTAFDB[16], LTDB[22], LTSTDB[23], MITDB[14], NSRDB[22], STDB[24] and SVDB[25]. Table 1 represents the number of patients, the measurement length of each patient, and the number of annotated heartbeats in the database (only AFDB was annotated by our QRS detection algorithm as a demonstration of clinical practice in the real world scenario, and the remaining databases were annotated by two doctors individually, where a third was invited to decide for a confronting annotation). We label this aggregated dataset by Full ECG DataBase collection (FDB). In finetuning, MITDB is used for training, and all other databases are used for testing. We label this collection of ECG benchmark databases, excluding MITDB, by Testing ECG DataBase collection (TDB), such that TDB = FDB - MITDB. The selected ECG benchmark databases were created for the following purposes:

- MITDB contains a variety of arrhythmia used in standards[26] and[27] for performance evaluation of ambulatory ECGs;
- AFDB and LTAFDB contain AFIB rhythm episodes;
- AHADB and CUDB contain different forms of ventricular arrhythmia;
- NSRDB and LTDB contain NSR;

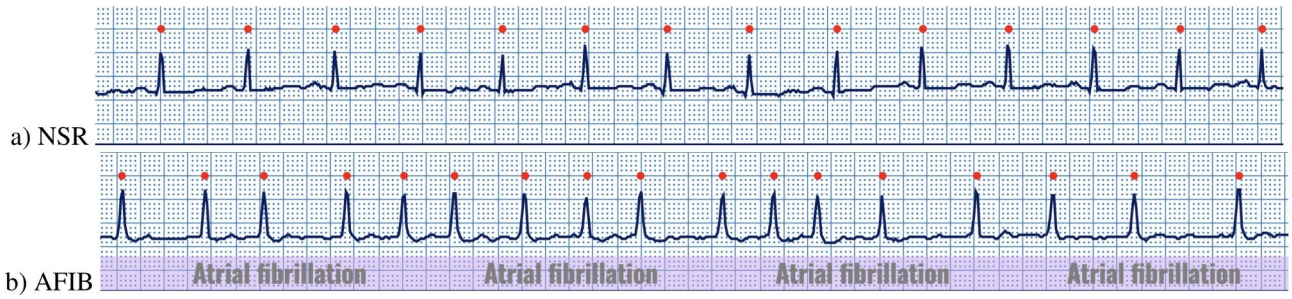

Fig. 1. ECG illustration of (**a**) NSR, a regular rhythm with patterns, and (**b**) AFIB rhythm, an irregular rhythm without patterns.

DB name	# Patients	Length	Original HBs	MF	Multiplied HBs
AFDB	25	10 h	1,069,956	4	4,279,824
AHADB	80	2 h	174,763	15	2,621,445
CUDB	35	1/7 h	19,476	50	973,800
EDB	90	2h	764,661	5	3,823,305
INCARTDB	32	1/2 h	175,907	20	3,518,140
LTAFDB	84	24-25 h	8,995,973	1	8,995,973
LTDB	7	14-24 h	667,816	6	4,006,896
LTSTDB	86	21-24 h	8,897,780	1	8,897,780
MITDB	48	1/2 h	107,656	40	4,306,240
NSRDB	18	23-25 h	1,724,412	2	3,448,824
STDB	28	2/9-9/8 h	76,172	40	3,046,880
SVDB	78	1/2 h	183,586	20	3,671,720

Table 1. ECG DB repetition counts to train the foundation model.

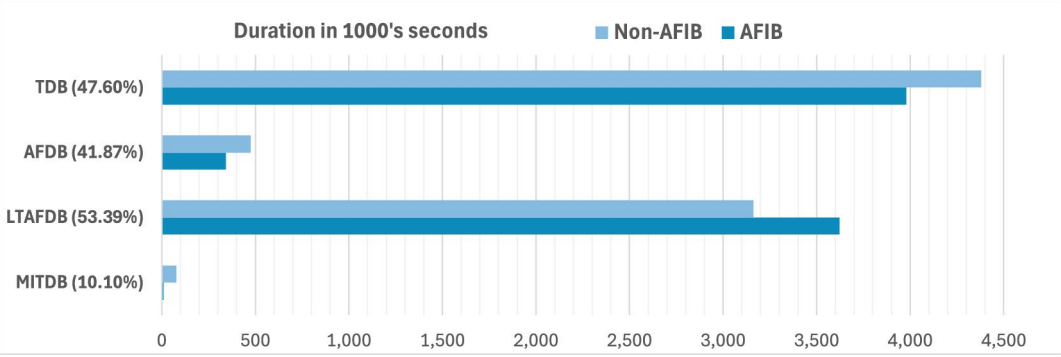

Fig. 2. Durations of AFIB and non-AFIB rhythm episodes in seconds.

- SVDB and INCARTDB contain individual ventricular and atrial/supraventricular beats with NSR as the underlying rhythm;
- EDB, STDB, and LTSTDB contain information on ST elevation and depression.Only three databases contain a reasonable portion of AFIB rhythm episodes presented in Fig. 2, and the others do not contain AFIB rhythm episodes or only a tiny portion (EDB 0.19%, QTDB 1.17%, and INCARTDB 2.67%). We chose a broader set of benchmark databases to train a foundation model to detect inherited rhythm patterns and finetune them for the specific AFIB classification task.

The number of heartbeats (HBs) in the ECG DBs greatly varies, such that LTAFDB and LTSTDB datasets include ECG recordings with over 8M heartbeats. The CUDB contains recordings with less than 20K heartbeats (Table 1). To avoid the influence of the extensive databases and getting lost as a minor particle in the vast data volume of large databases, we decided to include the exact content of some databases several times by a multiplication factor (MF) and match the volume of approximately 40 % compared to the LTAFDB and LTSTDB. In addition, we set a limit of the MF to 50. This multiplication approach optimizes the usefulness of highly valued subsets of data. Besides the number of heartbeats (HBs), Table 1 presents the MFs and the new multiplied number of HBs. The total number of HBs in FDB and TDB is 22,858,158 and 22,750,502, respectively. After applying the multiplication factor, the total number of beats in FDB gets 51,590,827.

MFs were not derived from a standardized algorithm. Still, they were selected through an approach to harmonize the quantitative disparities across the collected datasets and balance the differences in the number of heartbeats across datasets. The goal was to reduce the variance in the number of heartbeats each database contributed to the aggregate data corpus without excessive data duplication, which could lead to overfitting in predictive modeling or skew the statistical analysis results.

MITDB was also subject to a considerable multiplication factor. This decision was based on the proven quality of this dataset due to a diverse spectrum of arrhythmic conditions, including bigeminy and trigeminy arrhythmia, which are irregular rhythms but with a predefined pattern that may confuse an AFIB detector. For example, a bigeminy in a series of *RR* intervals (in ms) 800, 500, 820, 520, 810, 530, 840, 540 generates rhythm irregularity for which an AFIB detector based on photoplethysmography (PPG) may classify an AFIB rhythm since it does not analyze the beat types.

The foundation model training uses datasets in Table 1, particularly *RR* interval values and beat type information from 50M heartbeats. However, to finetune the foundation model for the AFIB detection binary

classification task, we used MITDB as a training dataset and all other datasets as a test set for evaluation purposes. MITDB is a golden standard for benchmarking ECG-based algorithms and models covering various arrhythmias. It ensures that the model is exposed to many different patterns, even irregular rhythms with patterns, such as bigeminy and trigeminy, to learn the correct AFIB rhythms. Due to the high volume and variety of AFIB rhythm sequences, the most common datasets for evaluating AFIB detectors are AFDB and LTAFDB.

To emphasize the class balance between the duration of AFIB rhythm sequences (positive class) and regular rhythm sequences, including other arrhythmias, we present the portion of positives concerning the total duration of episodes. In the feature engineering process, we excluded paced beats and fusion between normal and paced beats, intervals with higher noise that prevent QRS detection, ventricular fibrillation (VFIB), and ventricular flutter (VFL) rhythm episodes from the datasets since these rhythms are detected using different approaches. Figure 2 presents the duration of the AFIB (positives) and non-AFIB (negatives) rhythm sequences, the total duration (excluding the VFIB and VFL episodes) measured in seconds, and the portion of AFIB episodes expressed in percentages. In addition to testing the finetuned model on MITDB, AFDB, and LTAFDB, we include testing of other datasets that trained the foundation model, excluding MITDB, which was used to train the finetuning model, and expressing the total duration of TDB.

Although MITDB contains only 10.10 % of AFIB sequences, we use it to train the finetuning model since it includes various arrhythmia containing irregular rhythms with patterns that might confuse AFIB detectors, as analyzed in the Discussion Section.

Feature engineering

ECGs may provide a more complex view of the heartbeat morphology (Fig. 3) based on the characteristic QRS points, including P waves that represent atrial depolarization and precede the primary QRS complex or T waves that represent depolarization of the ventricles. Two crucial AFIB detection features in the ECG are the absence of P waves and irregular rhythm without patterns. Most detection approaches are based on evaluating the heartbeat regularity, analyzing the *RR* intervals as intervals between neighboring heartbeats and, more precisely, between R points from neighboring heartbeats, as presented in Fig. 3.

This research aims to develop a method to detect AFIB for single-channel wearable ECG sensors. We are aware that the position of the wearable sensor may make the P wave invisible or impossible to detect, especially for a physically active patient, where muscle noise generates higher levels. Therefore, we only used features from the QRS complex and corresponding RR intervals. In addition to this feature engineering approach, other research uses a series of digitized ECG samples or generates images of the ECGs to feed NNs or other DL methods.

We resampled the incoming ECG records to a unified sampling frequency of 125 Hz to ensure that all databases were analyzed the same way, using the same scale, since the analyzed datasets were sampled at 360, 250, 200, and 125 Hz. The extracted features identify the beat annotations by location and beat class. The *RR* interval, measured in milliseconds (*ms*), is essential for detecting irregular heart rhythms. Figure 4 presents the *RR* intervals on an ECG recording taken from an actual patient.

In addition, we extract another feature $dRR_i = RR_i - RR_{i-1}$ as the difference between successive *RR* intervals identified by $i = 2, 3, \ldots$. Our experiments showed that *dRR* provides much better information than the RR interval, also confirmed by other research[28].

The annotation files were combined with a sliding-window approach to create the training and test datasets systematically. This sliding window approach involved obtaining a sequence of N_{seq} consecutive heartbeats, which determine $N_{seq} - 1$ *RR* intervals, or $N_{seq} - 2$ *dRR* values. The sliding window approach generated all sequences of N_{seq} beats, avoiding beats that belong to VFIB or VFL arrhythmia. A sequence is labeled with the positive class (to belong to AFIB) according to the majority rule that classifies the sequence if most of the included beats belong to an AFIB sequence.

Development of the foundation model

Many papers have been published on detecting AFIB using various ML, DL, and NN approaches. This research uses an entirely new approach with transformers and LLM.

Our method (Fig. 5) uses the HuggingFace transformer library to provide textual input for RoBERTa as the Transformers NN architecture for building language models.

Figure 4 presents an ECG strip of an actual patient with identified R peaks. The corresponding RR intervals and *dRR* values are presented in *ms*. A character is associated with each consecutive *dRR* value mapped by the

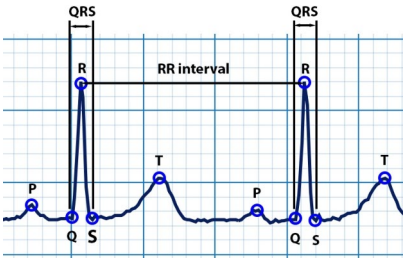

Fig. 3. Characteristic features of a heartbeat in ECG.

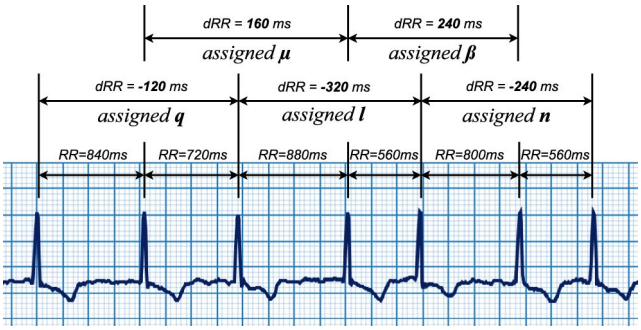

Fig. 4. ECG strip with annotated RR intervals (in ms), *dRR* values (in ms), and encoded characters (presented bold) using the mapping from Table 2. Text encoding of the given ECG interval is $q\mu l\beta n$.

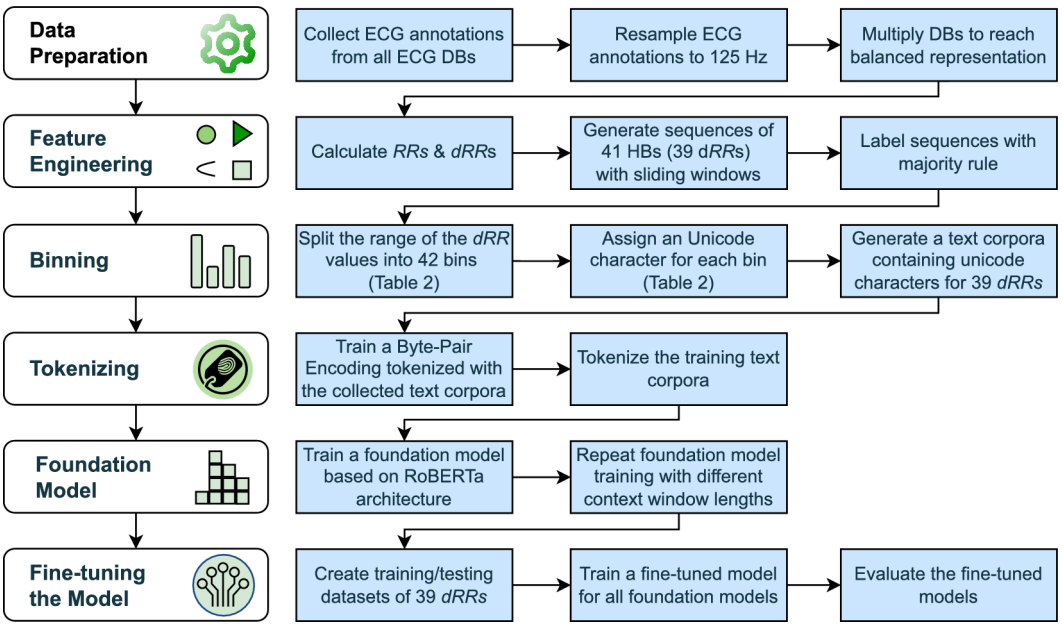

Fig. 5. Flowchart that systematically outlines the entire workflow from taking raw ECG annotations to finetuned AFIB classifier for sequences of 41 heartbeats.

encoding scheme from Table 2. For example, the sequence of RR intervals is 840, 720, 880, 560, 800, and 560, and the calculated *dRR* values sequence is -120, 160, -320, 240, and -240, and the associated character sequence is "q$\mu l\beta$n" using the encoding scheme.

The underlying algorithm to develop a foundation model (Fig. 5) consists of the following:

(1) Transforming ECG data into a text corpus for training the foundation model,
(2) Training a tokenizer based on the text corpus,
(3) Creating a foundation model training dataset with token IDs and
(4) Training the initial foundation model in a self-supervised manner.

Data transformation

We transform *dRR* values into a textual format by the HuggingFace transformer library[29]. Each numerical *dRR* value is converted to an alphabetical character (Table 2) by the following transformation methodology, which is a combination of equal-width and equal-populated bin strategy:

(a) A binning process was performed on the *dRR* values. First, we extracted the range between the 1st and 99th percentile of all *dRR* values. Then, this range was divided into two parts: one with negative and one with positive parts. Each part is split using the equal-width binning strategy, where each bin receives the same length, except for those bins approaching zero.
(b) A lower length is assigned to the bins close to zero to avoid being populated with too many *dRR* values,
(c) Additionally, two extra bins were created: one for values lower than the minimum of the 1st percentile and another for values higher than the maximum of the 99th percentile.

dRR interval (msec)	Mapped character	Unicode value
$(-\infty, -692.64)$	**a**	U+0061
$[-692.64, -656.16)$	**b**	U+0062
$[-656.16, -619.76)$	**c**	U+0063
$[-619.76, -583.36)$	**d**	U+0064
$[-583.36, -546.96)$	**e**	U+0064
$[-546.96, -510.48)$	**f**	U+0066
$[-510.48, -474.08)$	**g**	U+0067
$[-474.08, -437.68)$	**h**	U+0068
$[-437.68, -401.28)$	**i**	U+0069
$[-401.28, -364.80)$	**j**	U+006A
$[-364.80, -328.40)$	**k**	U+006B
$[-328.40, -292.00)$	**l**	U+006C
$[-292.00, -255.60)$	**m**	U+006D
$[-255.60, -219.20)$	**n**	U+006E
$[-219.20, -182.72)$	**o**	U+006F
$[-182.72, -146.32)$	**p**	U+0070
$[-146.32, -109.92)$	**q**	U+0071
$[-109.92, -73.52)$	**r**	U+0072
$[-73.52, -37.04)$	**s**	U+0073
$[-37.04, -0.64)$	**t**	U+0074
$[-0.64, 0.00)$	**u**	U+0075
[0.00, 0.42)	**v**	U+0076
[0.42, 31.76)	**w**	U+0077
[31.76, 63.2)	**x**	U+0078
[63.20, 94.56)	**y**	U+0079
[94.56, 125.92)	**z**	U+007A
[125.92, 157.28)	ª	U+00AA
[157.28, 188.64)	μ	U+00B5
[188.64, 220.00)	º	U+00BA
[220.00, 251.44)	β	U+00DF
[251.44, 282.80)	**à**	U+00E0
[282.80, 314.16)	**á**	U+00E1
[314.16, 345.52)	**â**	U+00E2
[345.52, 376.88)	**ã**	U+00E3
[376.88, 408.24)	**ä**	U+00E4
[408.24, 439.68)	**å**	U+00E5
[439.68, 471.04)	**æ**	U+00E6
[471.04, 502.40)	**ç**	U+00E7
[502.40, 533.76)	**è**	U+00E8
[533.76, 565.12)	**é**	U+00E9
[565.12, 596.56)	**ê**	U+00EA
[596.56, ∞)	**ë**	U+00EB

Table 2. Encoding scheme for transforming *dRR* values in characters..

(d) Each bin was assigned a unique lowercase Unicode letter, beginning with **a** for the first bin. Subsequent lowercase letters were assigned in ascending Unicode order until all bins had a corresponding character.After experimenting with various bin sizes, we used 40 inner and two outer bins in the binning strategy. Table 2 presents our encoding scheme (alphabet), such that any *dRR* value gets a unique character without obstacles to utilizing the Huggingface transformer library.

Training a Tokenizer

The text file of characters from converted *dRR* values is the input for training a tokenizer. Different strategies can be used for tokenization, each with its methodology and purpose. These include character-based tokenizers, where each character is treated as a separate token without further division, and subword-based tokenizers, such as Byte Pair Encoding (BPE)[30] and WordPiece[31]. Word-based tokenizers are also common in the literature[32]. Given the unique features of our text corpus, which lacks predefined words and sub-words and is not well suited for character-based tokenization due to its lack of data compression, Character-based tokenizers do not

compress data (one token per character), which is preferable since it allows the processing of bigger chunks of data simultaneously. Word-based tokenizers depend on well-established vocabularies, which exist for NLP. According to the state-of-the-art, we have yet to find evidence that they exist for ECG data, which leaves the BPE the only choice. BPE can effectively address the issues posed by the created text corpus, mainly it's capacity to create a more condensed representation of the data by combining characters into more significant tokens.

The tokenizer training process aims to construct a vocabulary of words as *n*-grams of varying sizes, which requires examining the frequency of *n*-grams of different sizes in the training dataset and determining the most effective way to divide the text into tokens. The objective is to create a vocabulary in which each word (sequence of characters) is assigned a token identifier, thus representing the dataset with the fewest tokens and the highest compression rate, significantly reducing the search space for further classification tasks.

Once the tokenizer is trained, it can assign tokens to a given interval of ECG data, which become inputs for the subsequent training procedure.

Creating a foundation model training dataset

Matching the size of a context window marked as *N* involves the creation of arrays of exactly *N* tokens according to the following procedure:

(a) Breaking down each text line into individual tokens to produce a single, long array of tokens for each annotation file, represented in a line in the intermediate text file.
(b) Applying a sliding-window technique with a single-step offset on each token array to generate a complete set of token arrays, each with a size of *N*.Upon completing this procedure, the resulting corpus comprises token arrays of size *N*, used as input in the foundation model training. This structured approach ensures a coherent and consistent dataset tailored to the specific requirements of the model-learning process.

Training the foundation model

GPT and other unidirectional models rely on the decoder part of the transformer architecture and are well suited for generative tasks, predicting the next most probable token. However, this research relies on bi-directional models based on an encoder and decoder to finetune trained foundation models for classification tasks with labeled data. We chose the RoBERTa bi-directional model since it excels in how it is taught (dynamic masking), considering other bi-directional models[33].

The model is trained with varying context window sizes (tokens). Table 3 presents the hyper-parameters used for generating the initial NN structure of the RoBERTa-based foundation model.

The original RoBERTa architecture specifies hidden layers size of 768, 12 hidden layers, 12 attention heads, and 3072 intermediate layers. However, considering a lower input size than the default size of 512 in three out of four experiments requires smaller values, and we have chosen those in Table 3. According to their best practices, the hyperparameters related to the training process (learning rate, optimizer, and other parameters) are preserved as provided by Huggingface.

Development of the finetuning classification model

Our method to develop an AFIB classifier is based on finetuning the developed foundation model (Fig. 5) by (1) creating the labeled training and testing datasets and (2) training the finetuning model.

Creating the labeled training and testing datasets

Finetuning the foundation model means training an AFIB classification model by corresponding training and testing datasets, which are constructed using tokens derived from the training dataset created for the foundation model. The tokens derived from 40 *dRR* values transformed into character-based tokens are organized into variable-size token arrays for computational processing. Subsequently, these arrays are normalized to a fixed size by including *PAD* tokens.

Splitting the training and testing datasets is crucial to avoid biased conclusions. In addition, since each database contains thousands of heartbeats from the same patient, it is essential to split the datasets by a *inter-patient approach*, such that all heartbeats from a patient will be included in one dataset, opposite to the *intra-*

Hyper-parameter name	Value
Vocabulary size	30522
Number of attention heads	8
Number of hidden layers	9
Hidden layers size	512
Intermediate layers size	2048
Learning rate	1×10^{-5}
Optimizer	AdamW
AdamW $\beta 1$	0.9
AdamW $\beta 2$	0.99
AdamW ϵ	1×10^{-6}

Table 3. Hyper-parameters to create the initial NN structure of the RoBERTa-based foundation model.

patient approach where the heartbeats from the same patient can be randomly shuffled and included in both the training and testing datasets. An inter-patient split develops more robust models and ensures that the model does not learn from data of the same patient in both the training and testing phases, helping to evaluate its generalization to new patients. We used the *cross-database inter-patient* evaluation, such that the training was done on MITDB and testing on other databases.

Fine-tuning the foundation model

The initial foundation model, originally trained self-supervised, was also finetuned using the Hugging Face Transformers library, leveraging various appropriate library software modules to ensure effective and efficient finetuning.

The model was trained in parallel batches on GPUs, dividing a dataset into smaller batches. During a one-iteration training step to process a batch, the GPU calculates the model's parameter updates (gradients) by (1) applying the optimization algorithm to each batch in parallel and (2) averaging them after synchronizing the batch tasks. An epoch refers to a complete pass through the entire training dataset. Since the dataset is divided into batches, one epoch consists of as many training steps as there are batches.

Evaluation methods

The most common way to evaluate a model, in general, is accuracy (ACC). However, the accuracy is not a reliable performance measure in cases where the dataset is unbalanced, with a significant difference in the number of positive and negative classes. The prevalence of imbalanced datasets in ECG recordings and other medical-related datasets has led to the dominant use of alternative performance metrics in research papers focusing on applying ML to medical data, such as recall or sensitivity (SEN), precision or positive predictive value (PPV), and F1 score, intended to more accurately reflect a model's ability to learn aspects essential to end-users and identify significant cases to them. Specificity (SPC), negative predictive value (NPV), and the F0 score give information about model behavior in measurements without AFIB.

In addition, since the F1 score measures predictive performance, we include the Matthews Correlation Coefficient (MCC)[34] to qualify the binary classification and avoid ignoring true negatives in calculating the F1 score. The Youden index (J)[35] is another measure used in this research that is equally relevant to true predictions.

The way to determine if positives and negatives are true or false can be via the number of included beats or the duration of the detected episode. Evaluation by the included fixed number of beats is relatively straightforward (*beat-based method*). Still, it needs to be more accurate since the duration of the episode is dependent on the heart rate. On the contrary, the *duration-based method*[36] measures the time of the correctly or wrongly detected episode (Fig. 6) and is used in corresponding standards EC57[27], and IEC 60601-2-47[26] to evaluate ambulatory ECGs for performance results of cardiac rhythm algorithms.

The sliding window approach determines labels for each sliding window. Furthermore, each heartbeat gets an AFIB or non-AFIB label, depending on most labels for all sliding windows containing the analyzed heartbeat. From a medical point of view, AFIB can be detected in an ECG interval if the irregularities in the RR durations last longer than 30 seconds[17,18]. We analyze labels of consecutive heartbeats to detect the beginning and end of the AFIB segments, remove intervals shorter than a predefined threshold, and join the surrounding non-AFIB or AFIB episode.

Results

All results are derived by coding the algorithm steps and executing the *transformers*[37] Python library. Training progress was recorded for training the foundation models and their finetuning, utilizing the *Weights and Biases* development platform[38]. In this research, we trained the tokenizer with a vocabulary size of 30522 as used in the standard RoBERTa model[7].

The value of N as the context window size in the training dataset varies between 41, 128, 256, and 512, specifying four experiments in our research while keeping all other hyperparameters constant.

The foundation and finetuning models are trained and tested on a high-performance computing environment comprising eight Tesla M60 GPUs with 8 GB of available VRAM.

Training foundation models

Progress in lowering the loss function during the training of all foundation models is presented in Fig. 7 within the first six epochs for the models with context window sizes of 41, 128, 256, and 512 tokens.

Training models with different context window sizes result in varying saturation speeds and steady loss values. Some models reach saturation quickly at a higher loss value, while others take longer to saturate but

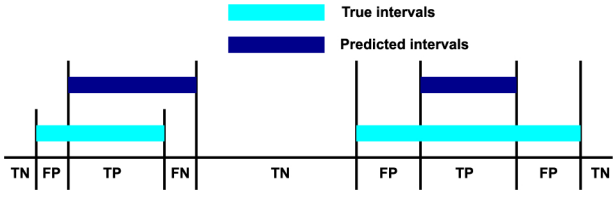

Fig. 6. Duration-based approach to calculate TP, TN, FP, and FN based on time instead of the count of included heartbeats.

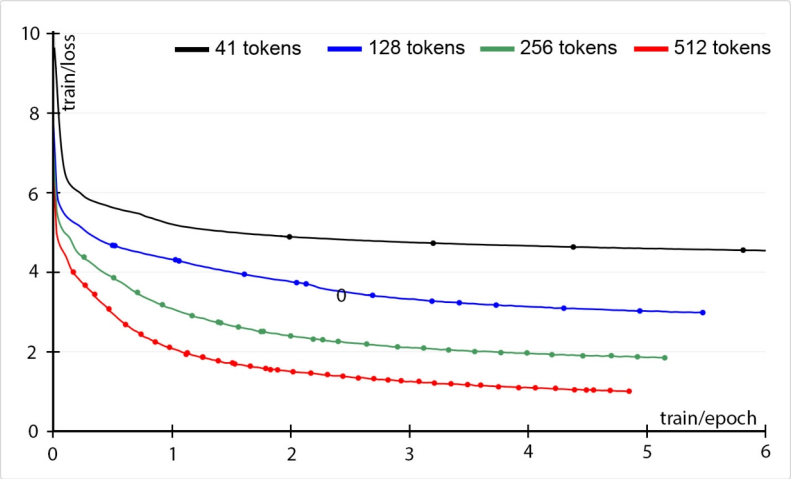

Fig. 7. Loss function versus epochs while training the foundation models with different context window sizes (41, 128, 256, and 512 tokens).

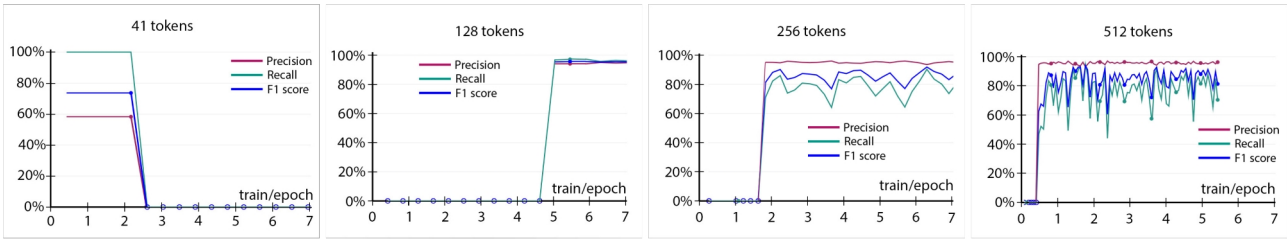

Fig. 8. Performance versus training epochs while finetuning the models with a context window of 41, 128, 256, and 512 tokens.

achieve a lower value. As there are no widely accepted criteria for automatically detecting the saturation areas (as a percentage of decrease in each new epoch), we applied a visual observation method of the loss function during the training process to detect regions where the curve seems to be flat and parallel to the epochs axis.

The achieved value of the loss function while training the model in a self-supervised way is not a reliable parameter to conclude how well the model is trained. When the original foundation models are finetuned, those saturated at lower loss values do not necessarily perform better when finetuned to downstream tasks than those saturated at higher loss values.

Figure 7 shows that the model with a smaller context window size of 41 tokens achieves saturation sooner than the models with a more significant context window. Conversely, models with bigger context window sizes show lower loss functions during training. The model with the most significant context window size, 512 tokens, achieves the lowest loss function values.

Fine-tuning models

Figure 8 presents the performance scores for finetuning foundation models with window sizes of 41, 128, 256, and 512 tokens correspondingly, calculated after each of the executed training steps on the training dataset (MITDB) and evaluation using the test dataset (LTAFDB). The foundation model with 41 tokens performs best in the first epoch, so it severely overfits each new epoch. Thus, a model trained with a short window size should be avoided due to its low-performance scores. Finetuning the foundation model with 128 tokens exhibits learning indicators from the fifth epoch onwards. Furthermore, there is no discernible evidence of overfitting occurring during the first 50 epochs. Moreover, there is no obvious indication that overfitting occurred during the training process. Applying finetuning of the foundation models with a context window size of 512 tokens shows signs of learning after the first part of the training epoch. Moreover, no obvious indication of overfitting during the training process has been found.

The performance scores for finetuned models with context window sizes of 41, 128, 256, and 512 tokens, trained on MITDB and evaluated on TDB, as a collection of all other databases excluding MITDB, show decreasing trends of SEN and increasing trends of PPV with increased context window size (Fig. 9). Sensitivity is higher for the models trained with shorter context window sizes and decreases with large context window sizes. PPV shows the opposite trend: it increases with increasing the context window size. F1 score performs optimally with a context window of 128 tokens; therefore, we claim that the model trained with that context window size is the most successful. Note that the 87.37 % F1 score of the model with 128 tokens is only 2.77 %, relatively higher

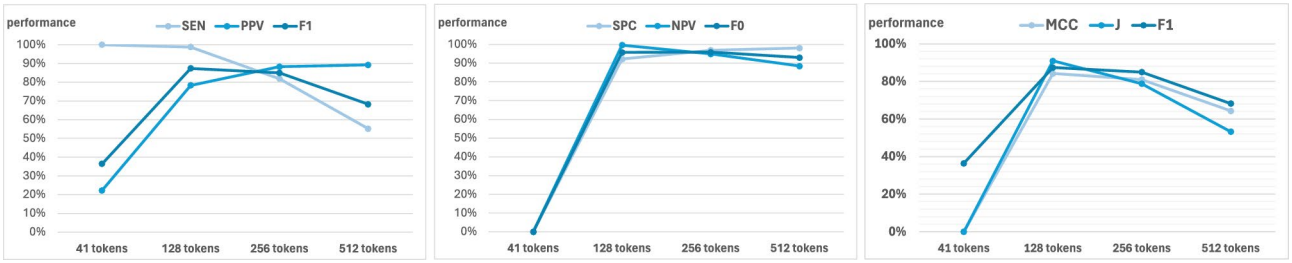

Fig. 9. Performance trends for models with a context window size of 41, 128, 256, and 512 tokens, evaluating TDB as a collection of ECG databases excluding MITDB used for training.

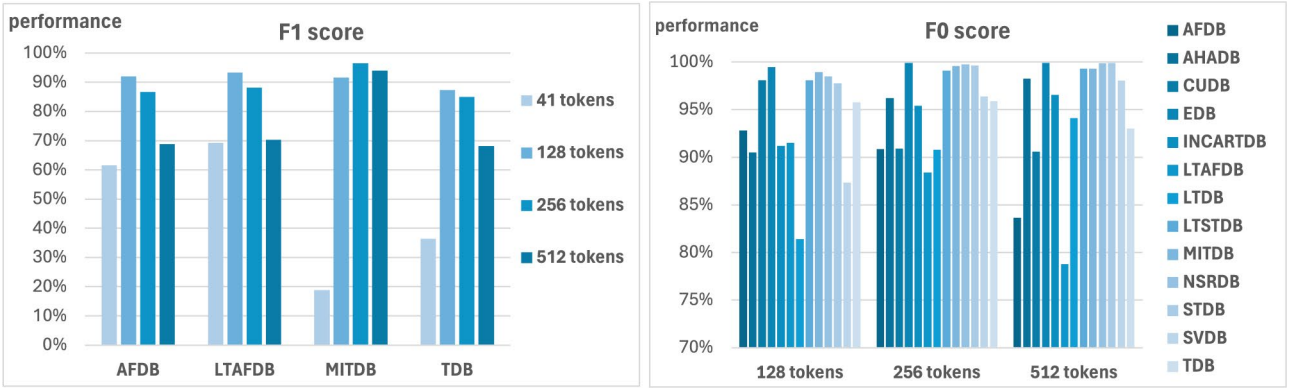

Fig. 10. F1 and F0 scores evaluating the models with a context window size of 41, 128, 256, and 512 tokens, trained on MITDB and evaluated on AFDB, LTAFDB, MITDB, and TDB (consisting of all databases excluding MITDB).

than the 84.95 % F1 score of the model with 256 tokens. Note that the 87.37 % F1 score of the model with 128 tokens is only 2.77 %, relatively higher than the 84.95 % F1 score of the model with 256 tokens.

The analysis of the performance scores on databases that do not contain AFIB rhythm episodes (Fig. 9) shows an increasing SPC trend and a decreasing NPV trend as the context window size increases. As the model with a window size of 41 tokens always classifies AFIB, the corresponding values are SPC=0.00 % without the possibility of calculating NPV and F0 (getting the value (N/A), such that this context size is insufficient to classify the rhythm episode correctly. The highest F0 score of 95.89 % is for the model with a context window size of 256 tokens, only 0.13 % more than the F0 score of 95.75 % for the model with a context window size of 128 tokens.

Figures 9 and 10 present a more detailed insight into evaluating the AFDB, LTAFDB, and MITDB databases that contain AFIB sequences, including the TDB, consisting of all databases excluding MITDB, which was used for training. As expected, MITDB achieves the best performance of 96.47 % F1 score as it was trained and evaluated on the same dataset with a finetuned model of 256 tokens. However, the model using 128 tokens performs best when evaluating other databases and is more robust. Model robustness is also confirmed by analyzing the MCC and J index values of 84.20 % and 90.94 % correspondingly (Fig. 9).

Databases that do not include AFIB sequences will result in PPV=0 and unavailability (*N*/*A*) to calculate the SEN and F1 scores. Therefore, Fig. 10 presents the F0 score to evaluate the model behavior on these databases referring to SPC and NPV. We observe that the model is robust and reaches F0 score values over 81.41 % with an average value of 95.45 % for the model with 128 tokens. Low values are observed for the LTDB and SVDB, which include sequences containing many atrial beats in the normal sinus rhythm, which confuses the AFIB detector model. In these cases, we conclude that a broader window context solves the differentiation of several atrial beats in normal sinus rhythm versus AFIB. However, the other databases reach high SPC and NPV values and an overall average value beyond 95.75 %, as observed by the results for MCC and J index values.

The experiments confirmed that the model with a context window size of 128 tokens offers a noticeable capability to capture mutual dependencies between the tokens inside the context window during the self-supervised training of the foundation model. This model is a good starting point for training a model specialized for AFIB detection with reliable predictions.

The model with a context window of 41 tokens wrongly detected everything as positive without detecting a negative class. The context window needs to be increased to capture mutual dependencies between the tokens inside this window during the training of the foundation model, which must be accurate enough to be a good foundation for successful finetuning. Overfitting happens after two epochs of training, and all the performance scores achieved by evaluating the model on the test data drop to zero (Fig. 8).

High-performance scores are evident for SPC and NPV for the model evaluation with a content window size of 256 tokens in different ECG databases, capturing mutual dependencies between the tokens inside the context window and self-supervised training of the foundation model. As it scores a slightly lower performance than the context window of 128 tokens, it is an alternative starting point for training a model specialized for AFIB detection. We observe that the performance scores for most databases are lower than those of the model trained with a context window size of 128 tokens, concluding that context 256 is too broad and captures irrelevant information or noise, preventing correct detection.

Although evaluating the finetuned foundation model with a context width of 512 tokens shows high performance for SPC and NPV, it is not viable since it reveals lower performance than the models with a context window size of 128 and 256 tokens. This model does not capture the mutual dependencies between the tokens inside the whole context window; thus, among two distance tokens, one generates noise for the other and vice versa. We found that a long context window is unsuitable for training foundation models with ECG databases due to the specific characteristics of this kind of data. Our findings show that patterns within extended ECG sequences may not manifest in a manner recognizable by transformers' architecture.

Performance scores between the beat-based and duration-based evaluations show discrepancies. Figure 11 presents the minimum, maximum, and average absolute differences between the beat and duration-based evaluations. As expected, the performance scores calculated using the duration-based method are lower than the beat-based method. Models with 256 and 512 tokens reveal negative absolute differences for AFDB and LTAFDB, meaning that the duration-based method scores higher F1 values than the beat-based method. However, the overall behavior is the opposite (Fig. 9).

The voting mechanism that decides whether a single heartbeat belongs to an AFIB episode leaves the possibility of making bad decisions about the beginning and end markings of AFIB and non-AFIB episodes. Removing short AFIB or non-AFIB segments with the moving-average approach affects the overall accuracy even more. Both factors unpredictably affect the increase or decrease in the accuracy of the duration-based measured F1 scores. Therefore, we observe no correlation between the duration-based and beat-based measured F1 scores and rely on the experimental results.

Discussion

State-of-the-art analysis

Our latest analysis of the state of the art shows no published approach for AFIB detection using transformers. Existing research papers on detecting AFIB in ECG recordings differ in feature engineering, ML technologies/algorithms, training, and testing data splits.

Feature engineering for AFIB detection

Several systematic reviews of AFIB detection methods analyze related work using various methods[2–4]. The sensing method of heartbeat locations can be based on:

- *PPG*, such as those in heart rate monitors[39], identifying only the preceding *RR* interval without identifying ventricular beats,
- *ECG*, used in monitors such as implantable loop recorders, Holter, and wearable patches monitors, identifying both the beat type and preceding *RR* interval[3].Researchers calculate various statistical features from R-peaks locations and RR intervals to train old-fashioned ML models and handle the data in multiple ways. Some use measured millivolt values to train 1-D convolutional NNs or transform them first using advanced signal processing techniques. In contrast, others draw 2-D images of the ECG recordings and use well-established 2-D CNN training technology. With the rise of Transformers, several researchers have used them to generate embeddings for pieces of ECG signals and use them as input in other ML techniques.

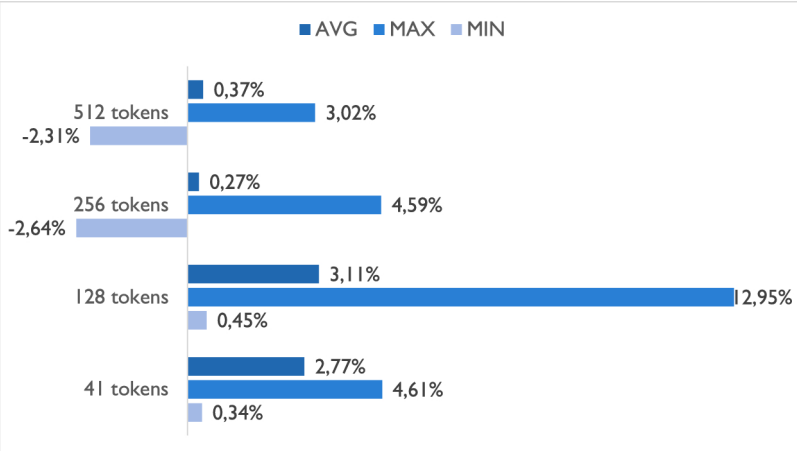

Fig. 11. Performance score Differences between beat and duration-based evaluations.

Shannon entropy[40] is mainly used as an indicator of regularity in signal processing-based algorithms[41]. Additional research methods include the turning point ratio[42], in combination with the root mean square of successive RR differences, as the distance between two successive *RR* intervals, and the Shannon entropy, or a set of specific characteristics[43]. Others used a combination of time-varying coherence functions[44] or analyzing the time-varying transfer functions, and Shannon entropy[45].

Alternatively, instead of analyzing the *RR* intervals, some authors use *dRR* intervals (as in our approach) or their combination. One of the influential papers on AFIB detection was published by Tateno and Glass[46], who proposed the coefficient of variation as a feature of RRI and *dRRI*. For example, Lian et al.[28] applied a threshold over many empty cells in an image constructed by *RR* distances (x-axes) and *dRR* values (y-axes) divided into a grid.

Transformers in ECG processing

Our state-of-the-art analysis showed that no paper uses the approach introduced in this paper. Only a few articles discuss the use of transformers to implement automated ECG analysis, where only Zhou et al.[47] used unsupervised pre-training and labeled finetuning methodology, and the others use training methodology with finetuning randomly initialized models. The approaches by different researchers vary in the training data volume, such that finetuning the models based on transformer architecture does not rely on preceding self-supervised training with a large ECG recording corpus. Most researchers use only one benchmark for finetuning (training) and testing, which may introduce biased conclusions. Most use only the decoder part of the complete transformer architecture, and some utilize CNNs to generate features from the raw ECG signal. No published approach uses tokenization, and the input in their models is a series of ECG values or embeddings extracted using CNNs. A state-of-the-art analysis shows no evidence that the transformer approaches used by other authors perform well on NLP tasks compared to the well-known models that provided the performance that made them famous. Our approach uses self-supervised pre-training with massive ECG recordings collected from 12 ECG benchmark databases. In addition, the finetuning and evaluation use different ECG benchmark datasets.

Zhao[48] reviews nine transformer-based DL models in cardiovascular disease detection and automatic ECG analysis. Three models use ECGs with 12 leads measured with stationary ambulatory equipment, and the rest are based on a single lead ECG signal, mainly based on short-term ECG recordings. All these analyzed approaches utilize CNN or RNN DL models for feature extraction with lightweight transformers, contrary to our approach based only on transformer architecture utilizing the full power of a bidirectional approach with multiple attention layers. In addition, the author does not specify the train/test data splits nor disclose whether data from the same patient (recording) is used for training and evaluation.

A classification framework, introduced by Atiea and Adel[49], analyzes the raw ECG signal in conjunction with the durations of the *RR* intervals to classify five primary classes (NORM, HYP, MI, STTC, CD) and fourteen classes based on the SCP-ECG standard[50]. The authors utilize a 1D CNN to acquire embedded ECG data and feed a transformer-based NN with a context window size of 512 tokens. The model is trained on the PTB-XL dataset[51] that contains 10-second ECG recordings from 12-lead ambulatory measurements. In addition, the testing uses the same dataset evaluation with a seven-fold cross-validation approach. The findings demonstrate that the suggested model achieves exceptional accuracy and an F1 score, averaging 99.86 % and 99.85 %, respectively. In medicine, the N cross-validation approach on the same dataset results in biased conclusions and the developed models with small performance on ECG recordings from untrained patients. The analyzed approach does not comply with the gold standard based on 30-second ECG strips to detect AFIB, as we use it in our approach. In addition, the differences in our approach include (1) N cross-validation instead of inter-patient evaluation with different datasets, (2) no unsupervised pre-training and finetuning, 3) 12-lead ECG data from 10-second recordings instead of one lead ECG from longer ECG strips, 4) classification of multiple classes, instead of binary classification of AFIB.

Chao et al.[52] make another attempt to use CNN-based ECG signal embeddings to feed a Transformer-based NN. They worked with ECG data acquired from a cardiology challenge[53], collected from 11 hospitals, and covered 6877 individuals. Using a 9/1 train/test data split, they achieved an F1 score of 78.6 % to classify AFIB and other arrhythmia types. Their model passes 12-lead ECG to CNN to extract features to finetune the initialized transformer composed by the encoder part only, without prior training. In contrast, our model uses one lead ECG and pre-trains a foundation model in an unsupervised manner. The training objective is classifying nine arrhythmia classes instead of just AFIB, as in our approach.

DistilBERT[54] is a starting point to transfer learning of ECG data collected from the MITDB, followed by finetuning for a downstream sequence classification task[55]. The analyzed sequence is an ECG interval with 254 digitalized samples left and right from the R peak. It classifies the heartbeat into one of four types, which is different from the main focus of this research, which is to detect AFIB as a rhythm consisting of several heartbeats. The authors utilize a pre-trained model for NLP tasks, with texts written in English, suggesting that the structure of natural languages is similar to that of heartbeat classes. However, transfer learning methodology applied to natural languages is unsuitable for any task related to data from the ECG domain. The published results of an F1 score of 99 % is a great result, likely attributed to the 8:1:1 split of training validation and testing. This high score may also result from using heartbeat pattern data from the same patient across all phases, which is opposite to our robust approach to using different ECG benchmark databases.

Zhou et al.[47] approach to the classification of ECG diagnosis by end-to-end training a Transformer-based model with the objective of a mask language model (MLM)[56], using the Fuwai Hospital of the Chinese Academy of Medical Sciences dataset consisting of 220,251 ECG recordings from 173.951 adult patients, and the PhysioNet / CinC Challenge dataset which contains 88.253 12-lead ECG recordings. The ECG DB of choice for evaluation was the PTB-XL ECG DB[51]. They experimented with architectures with different numbers of elements for the Transformers NN, and the best F1 score achieved was 78 %. The authors use a complete pipeline with proper

pre-training and finetuning phases using 12-lead ECG recordings lasting 10 seconds, while we use one-lead ECGs lasting 30 seconds to comply with the gold rule for detecting AFIB. The authors used a 10-fold train-test split, opposite to an inter-patient split used in our approach. Two experiments pre-train and finetune the same dataset, and one experiment different datasets for training (PCinC) and finetuning (PTB-XL). In contrast, our experiments use several datasets for training the foundation model and other datasets for finetuning and evaluation. Their approach results in a multilabel classification of arrhythmia, publishing the achieved macro F1 scores as the unweighted mean of the per-class F1 scores instead of the binary classification of AFIB in our approach.

Combining a vision transformer structure with deformable attention in conjunction with CNNs with separable depthwise convolution is an approach to classify eight cardiac arrhythmia types proposed by Dong et al.[57], training a model with data from the China Physiological Signal Challenge 2018 (CPSC-2018)[53] consisting of 6877 12-lead ECG recordings collected from 11 hospitals, instead of one lead ECGs used in our research. The authors achieved an F1 score of 82.9 % as the average of F1 scores for each detected arrhythmia. Opposite to our approach, the authors use CNN for feature engineering and do not pre-train the transformer foundation model. In addition, instead of an inter-patient approach in our research, they evaluated the model with 10-fold cross-validation for the multilabel classification task.

Comparison with other authors

Conducting a meaningful comparative analysis of various academic and industrial research poses a significant challenge. The results of these research efforts materialize in the form of algorithms, heuristics, and ML models designed to detect atrial fibrillation (AFIB), which is the central focus of this study. These tools answer whether AFIB is present or absent in specific segments of ECG recordings. Achieving a robust evaluation of AFIB detection methodologies and enabling a meaningful comparison requires consistency in training and testing datasets, as well as uniformity in the format of input data. Unfortunately, this necessary uniformity is currently lacking. However, to advance in this undertaking, some level of assessment becomes essential, though only partially precise.

It is important to note that splitting training and test data significantly impacts achieving high-performance results. The most reliable performance scores are obtained using different datasets for training and testing. This method guarantees an understanding of the potential performance metrics that can be reached by applying the trained model to unknown data collected in various situations. These performance scores are credible but are mainly the lowest among those reported in the research literature.

An alternative approach to establishing the train/test split is to divide all ECG recordings according to individual patients. Ultimately, the most effective approach to mathematically valid and exceptionally high-performance scores is to divide the train data set by random shuffling of all training examples gathered from all patients from all training databases when there are several of them. This approach typically yields performance scores higher than 99 %. However, this method is susceptible to a systematic error of data leakage. The model was trained and tested using data from the same patient, which resulted in specific learning patterns. Consequently, while the performance scores are high, the model's robustness is significantly compromised as it cannot accurately predict how it will perform, exposing it to unseen data from other patients.

The essential information about train/test splits that significantly affect the performance scores. Our evaluation focuses on the *cross-database inter-patient* method, where the train/test split uses different databases and patients for training and testing. Many authors use cross-validation, which differs from our approach since it uses samples from the same patients for training and testing.

Comparing the same evaluation model

We found only two papers on AFIB detection using the inter-patient evaluation.

In the developed LSTM model by Pereira et al.[58], the number of 10-second samples does not correspond to the declared values in the provided table, and we assume that the authors analyze only clean segments that contain only AFIB or non-AFIB segments. In a real-world scenario, as in our case, 10-second segments can be mixed using a sliding window approach. Also, the authors do not use the duration-based evaluation recommended by the corresponding standards IEC 60601-2-47[26] or EC57[27]. Their AFDB+ database uses a sliding window approach to develop 10-second segments, where each new window is displaced for two beats. The authors divided the histogram into three bins of RR intervals, where the central bin corresponds to the RR intervals close to the mean value, and the number of RR intervals smaller than 90% of the mean RR interval or greater than 110% specify the left and right bins correspondingly. The authors apply this binning strategy to *dRR* and supply the model with four features. Following the inter-patient paradigm, the performance was assessed through a ten-fold cross-validation methodology, such that the ECG signal segments from the same individual did not appear in both the training and test subsets (20% of the training subset for validation).

Jahan et al.[59] have analyzed the following seven RR features (mean of RRI, standard deviation of RRI, root mean square of successive differences, normalized root mean square of successive differences, Shannon entropy, mean and median absolute deviation) and experimented with six ML algorithms (Support vector machines, Decision tree, Random forest, Stacking classifier, Extreme gradient boosting, and Adaptive boosting) testing sequences of 10, 20 and 60 beats. Their evaluation approach matches ours using the intra-subject scheme. They concluded that almost all of the studies with intra-subject schemes achieved sensitivity and specificity higher than 95% and 89%, respectively, while these values are relatively lower for inter-subject studies. However, the robust inter-patient method is realistic since AFIB rhythm episodes are evaluated on unseen test sets, and the method can be efficiently applied in clinical settings.

Different testing and evaluation methods make it impossible to compare with other authors' performance results directly. Most approaches use samples of clean segments with a constant duration so the samples do not

overlap and contain only beats of one class (AFIB or non-AFIB). We use a sliding window approach to fit a real-world scenario where the incoming stream of ECG is interpreted as it comes. This strategy can detect the onset and offset of AFIB. The highest challenge is the classification of samples containing beats in AFIB and non-AFIB rhythm episodes, and their label is determined according to the majority rule. In addition, this strategy complies with medical standards for the performance of ambulatory ECGs IEC 60601-2-47[26] and EC57[27].

Most researchers use only sensitivity, specificity, and accuracy as performance measures. However, it would be good if AFDB and LTAFDB realized the testing. In a real-world scenario, the prevalence of AFIB is much smaller, and the F1 score is a much better indicator than accuracy. Specificity indicates how much the detector made a wrong estimation if there is no AFIB. When the number of AFIB segments is much smaller, then this information is misleading since false negatives may dominate over the true positives and decrease the positive predictive value. In this case, the sensitivity, specificity, and accuracy will be high, but the overall performance with an enormous number of false negatives.

Presenting the false positive rate (FPR) as a performance indicator is obsolete if the paper presents the specificity since FPR=1-SPC.

Some influential signal processing algorithms are presented in Table 4. Their corresponding thresholds were tuned to fit the initial databases, and these algorithms can be tested only with cross-database inter-patient evaluation. Unfortunately, this kind of evaluation is rare. Almost a similar case is with the ML and DL algorithms, which train and test with samples from the same patients. In most of the analyzed literature, the training and testing data splits were not indicated, raising doubt that the test was done with samples from the same patient.

Comparison with multiple classifiers is only possible if the relevant performance measures, including sensitivity and F1 score, are presented for classifying AFIB. Comparing only overall accuracy results in higher values from prevalent classes overrides the relevance of smaller arrhythmia.

Our evaluation method uses the ViweECG QRS detector to detect real-time continuous ECG streams. For the incoming stream, we detect the R peak with a high F1 score of 99.90% and then continue analyzing the RR intervals. We know that the QRS detector may detect a false positive (FP) or false negative (FN), especially when muscle noise is high with many artifacts of a physically active person with a wearable ECG sensor. To avoid corruption of the AFIB detector, we integrated filters to deal with smaller noise amounts of noise (SNR<12db) and detectors of high noise signal levels, making the signal uninterpretable. In addition, we trained the foundation model from ten annotated ECG databases and one ECG database (AFDB) annotated from our ViewECg QRS detector. The finetuning model to classify AFIB was realized with an already annotated ECG database (MITDB). During training, we generated overlapping samples using a sliding window approach to train the foundation model to recognize the context of the rhythm pattern.

Although we analyze the Physionet Challenge 2017 approaches to recognize their scientific contribution, we do not directly compare the achieved performance since they needed more space to fit their model to test data.

Approach	Training	SEN (%)	SPC (%)	PPV (%)	F1 (%)
AFDB					
Pereira et al.[58]	AFDB	70.97	73.70		
Pereira et al.[58]	AFDB+	91.53	**91.08**	N/A	N/A
Jahan et al.[59]	AFDB 20b	87.58	89.27	87.78	87.99
Jahan et al.[59]	AFDB 60b	91.92	87.45	**96.21**	**94.02**
This work	MITDB	**97.84**	88.07	86.76	91.97
MITDB					
Jahan et al.[59]	AFDB 20b	85.67	81.25	90.85	88.18
Jahan et al.[59]	AFDB 60b	83.62	95.24	**92.89**	88.01
This work	MITDB	**98.81**	**98.02**	85.31	**91.57**
LTAFDB					
Pereira et al.[58]	AFDB	74.21	67.36	N/A	N/A
Pereira et al.[58]	AFDB+	89.17	87.03	N/A	N/A
Jahan et al.[59]	AFDB 20b	86.45	81.57	**87.06**	86.75
Jahan et al.[59]	AFDB 60b	85.13	82.53	85.65	85.39
This work	MITDB	**98.90**	**88.36**	85.39	**93.33**
NSRDB					
Pereira et al.[58]	AFDB	N/A	82.95	N/A	N/A
Pereira et al.[58]	AFDB+	N/A	93.21	N/A	N/A
Jahan et al.[59]	AFDB	N/A	96.42	N/A	N/A
This work	MITDB	N/A	**97.05**	N/A	N/A
TDB					
This work	MITDB	98.76	92.18	78.34	87.37

Table 4. Performance scores for AFIB detection with RR intervals and cross-database inter-patient evaluation method. Significant values are in bold.

Table 4 compares the AFIB detection approaches that publish results with the cross-database inter-patient method, meaning that the model was trained with records from one database and tested with the records from another unseen database.

Note that direct comparison to other approaches is impossible since our finetuned model trained a foundation model with the whole MITDB database. The foundation model captured the token context from 12 ECG databases, while the other models were trained with different ML and DL approaches, mostly on AFDB only. Therefore, we compare the results from the cross-database inter-patient method, such as testing the LTAFDB.

Analyzing the performance achieved on LTAFDB, we found that our method outperforms the others, primarily by achieving the highest sensitivity result. Although our finetuned model is trained on MITDB, it achieves the highest sensitivity when evaluating the other databases. This performance is due to Bigeminy, Trigemeny, and similar arrhythmia in the training dataset and the foundation model's capability to process tokens and extract common relations, representing rhythm patterns.

Due to the smaller sensitivity values, Jahan et al.[59] achieve the best positive predictive values. In medicine, doctors prefer better sensitivity for the price of lower positive predictivity; in other words, it is much better to have more FPs than FNs.

Specificity is an important measure expressing a smaller number of FNs. The analysis shows that our model is better than the others for the cross-database inter-patient method. Only the model of Pereira et al.[58] trained by a sliding window approach (with a step of two beats) performs better, as it is tested on the same AFDB dataset it was trained for.

AFDB+ is most similar to our approach since it uses the sliding window approach to generate the samples. However, in the testing phase, Pereira et al.[58] use only clean samples with a label of AFIB or non-AFIB, and we evaluate even the mixed segments and decide the label with the majority rule.

We must acknowledge that the performance results of Pereira et al.[58] are only for short 10-second segments, while our model is on 41 heartbeats (approximately 30 seconds). In contrast, Jahan et al.[59] experimented with 10, 20, and 60 beats (approximately 8, 16, and 50 seconds, depending on the heart rate).

In addition, we use the duration-based evaluation method, while the others use the sampling method to test only pure AFIB and non-AFIB segments.

Limitations

The application of this model is technically limited due to certain assumptions made before developing the model:

- *ECG signal lead selection*: This research primarily uses the first channel of the two lead measurements, mainly corresponding to the MLII lead, where the R peak in the QRS is positive. However, since several records in the datasets are from other ECG leads, and because the ViewECG QRS detector has proven to perform with an F1 score of 99.90%, the solution may also perform similarly without this limitation.
- *Noise-free ECG signal* is expected as input, or an ECG signal with signal-to-noise ratio (SNR>12db). In such cases, conventional signal processing algorithms and filters that do not modify the signal morphology integrated into a noise elimination module can eliminate noise and fix baseline wander in applications for wearable ECG sensors. It is recommended that the system implement a detection module to process the input, which detects high noise levels and classifies an uninterpretable signal to avoid further errors. Only ECG signals with slight noise levels can be further processed for satisfactory QRS detection and provide sufficient annotations for the model input.
- *No VFIB and VFL intervals* are provided for further processing the model's input since the QRS detector can detect irregular rhythms and output an AFIB detection. Usually, each ECG detection and classifier system integrates such a module after the noise detection and noise elimination modules.
- *Distinction from Atrial Flutter (AFL)* is essential since this research detects AFIB, where the atria beat irregularly. The developed model does not address atrial flutter (AFL), in which the atria beat regularly but faster than usual and more often than the ventricles.Next, we present the lessons learned about developing a heart-based language based on transformers, compare the achieved results to detect AFIB with other research, and elaborate on this research's main contribution.

Lessons learned

1. *Correlation of training performance between foundation and finetuning models*. Table 5 and Fig. 9 present the training duration measured in epochs and the achieved performance of the foundation model and finetuning. The loss achieved while training the foundation models, calculated as cross-entropy between input and

Context window	41	128	256	512
Training foundation models				
Epochs	10	5.47	5.15	4.85
Final loss	4.46	1.98	1.84	0.99
Fine-tuning trained foundation models				
Epochs	10	50	33.54	5.43

Table 5. Indicators for training the foundation models and finetuning.

output, is lower for the models trained with more oversized context windows. However, it is not correlated with the performance scores achieved by evaluating the finetuned models, meaning that the quality of the foundation model cannot be estimated by the value of the achieved training loss, and models with higher loss can perform better when finetuned.

2. *Speed of the loss function drop*. The larger the context window size, the faster the loss function drops during training for all developed models with 41, 128, 256, and 512 context window sizes (Fig. 7).
3. *Impact of the loss value*. Foundation models with lower loss achieved during the training do not necessarily perform better than those with higher loss function when finetuned to downstream tasks (Table 5 and Fig. 9). When trained for binary classification, the model with 128 shows superior performance scores compared to other tested models. Shorter sequences needed padding with tokens at the end to reach the training input size.
4. *Small context window size*. The small context window in our experiments was 41, which cannot provide signs of learning capabilities during the finetuning process despite the observed drop of the loss function while training the foundation model as observed in Figs. 8 and 9.
5. *Large context window size*. A more significant context window size does not necessarily provide better-finetuned models. In our case, a model with a context window size of 128 performs better than those with 256 tokens and even better than those with 512 tokens (Fig. 9).
6. *Impact of the evaluation method*. The discrepancies between the measurement of beat-based and duration-based performance scores, the F1 score, in particular, vary. In some experiments, the beat-based method outperforms the duration-based method, and the opposite is true in other cases. F1 scores for all DBs tested with the finetuned model with 128 tokens are higher when measured with the beat-based method than when calculated using the duration-based method for the best-performing model. However, concerning the models with 256 and 512 tokens, for some DBs, the F1 score is higher when measured using the duration-based method.
7. *Model performance behavior*. The finetuned model has been trained to detect specific AFIB and non-AFIB patterns, using a relatively small number of similar patterns to trigger accurate classification. AFIB is defined as an irregular rhythm over a 30-second interval. Assuming an average heart rate of 72 BPM, we use 41 heartbeats and 39 *dRR* intervals. The total number of samples is enormous since the training samples are generated using a sliding window approach. Although the model performs excellently on recordings with heavy irregularity, it makes mistakes in cases where the similarity with the learned patterns is thin or not that hard. We conclude that the learned features from the training data do not have enough corresponding examples in the test set, resulting in fewer opportunities for the model to apply its AFib detection capability.

Error analysis

The model makes mistakes in cases where the similarity with the learned patterns is thin or not that hard. Usually, these cases are borderline cases of the decision criteria. An example of an FP error is illustrated in Fig. 12. The occurrence of one premature atrial beat (A) and eight premature ventricular beats (V) with two short episodes of ventricular bigeminy introduced an irregularity that confused the detector. This resulted in records 200 and 228, with 71 and 20 episodes of bigeminy reaching FPR values of 2.46% and 3.34%, respectively. Trigeminy is another factor influencing the higher rates of 10.78% and 4.45%, respectively, for records 201 and 219. Ventricular tachycardia in record 203 also impacts the higher FPR. A higher number of V beats in records 200, 201, 203, and 228 in the range over 10% resulted in reaching a higher FPR. Similarly, records 222 and 232 had a higher number of A beats, with values of 8.42% and 77.91%, respectively, causing the highest FPR of 17.57% and 18.45%.

Figure 13 illustrates an FN error since the compensation pauses after the ventricular beats happened to be equal or belong to the same interval and map to the same character (Table 2) and reveal a minor irregularity degree. FNs in MITDB records were present only in records 219 and 222 with high FNR rates of 2.32% and 3.25% due to 2.97% V beats in 219 and 8.42% A beats in 222. Table 4 shows that our model reaches a high sensitivity of 98.81% evaluating the MITDB, corresponding to a low FNR of 0.12% for the overall MITDB. In contrast, the FPR rate of 1.77% corresponds to a lower PPV of 85.31%.

Figure 14 presents the error rates evaluating the analyzed ECG benchmark databases. When evaluating the TDB, the model results in an FPR of 6.08% and FNR of 0.28%. Similarly, when assessing the FDB, the model results in an FPR of 6.06% and an FNR of 0.28%. The performance of FDB and TDB differs by 0.02% for FPR, and differences in FNR are neglected due to the small impact of MITDB on the overall number of heartbeats in the evaluation.

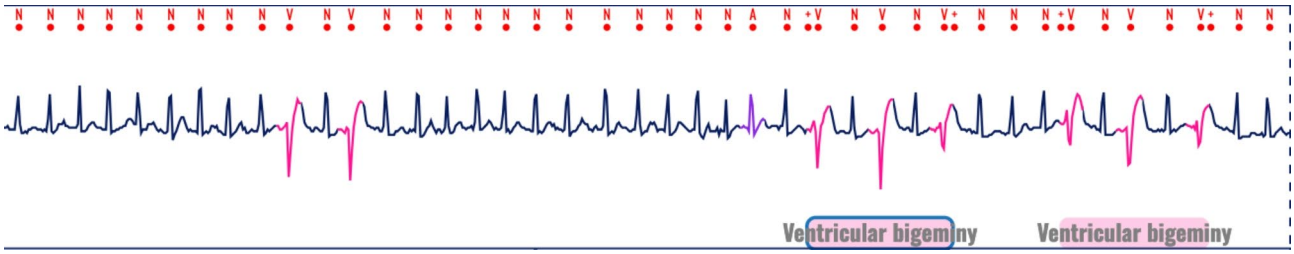

Fig. 12. An example of an FP error (MITDB record 200).

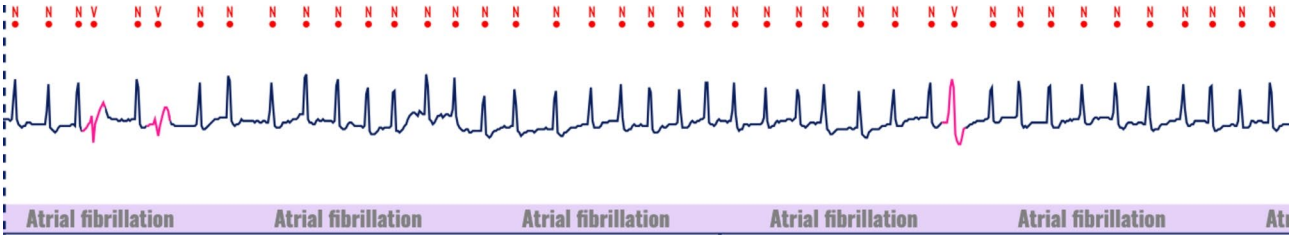

Fig. 13. An example of a false negative error (MITDB record 210).

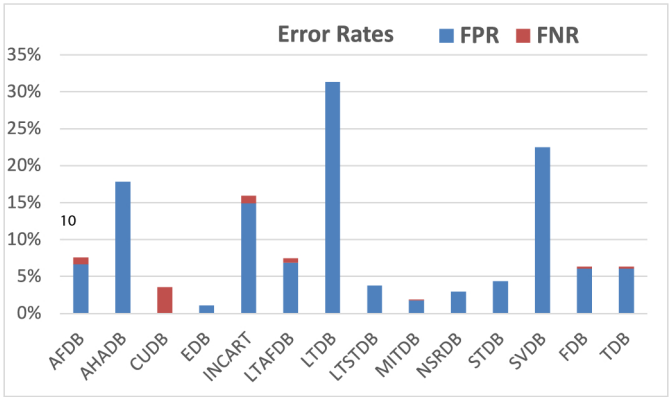

Fig. 14. Error rates of our model for all analyzed ECG benchmark databases, including the FDB and TDB.

The in-depth analysis of the model performance shows higher FNR for CUDB, which contains only patients with ventricular arrhythmia. FNR up to 1% is found in AFDB and INCARTDB, and 0.58% in LTAFDB due to the higher number of V beats with compensation pauses after V beats that happened to be mapped in the same character in the encoding scheme (Table 2).

High FPR rates beyond the 6.06% average (Fig. 14) appear in LTDB, SVDB, AHADB, INCARTDB, LTAFDB, and AFDB due to the higher number of V beats and episodes of bigeminy, trigeminy, and ventricular tachycardia.

Reaching only 0.28% FNR means that the model is extremely sensitive and captures all relevant AFIB episodes with a minor number of missed detections, which is valuable in medicine. The 6.08% FPR means that the model needs more training to be more precise.

We conclude that the learned features from the training data need more corresponding examples in the test set to result in a model with increased AFIB detection performance.

Contribution

The research demonstrated in this paper showed that it is possible to train a foundation model with a high volume of ECG data (containing 22M heartbeat annotations) in a self-supervised way and then finetune it for the downstream task, mainly to detect AFIB sequences with a much smaller portion of labeled training data (MITDB containing 100K heartbeat annotations). The best-fine-tuned model outperformed most currently trained models documented in scientific papers using the inter-patient and duration-based evaluation. The model robustness is confirmed by analyzing the achieved performance measures, including the F1 score, F0 score, MCC, and J indexes. The results indicated the effectiveness of this approach and suggested that the experiment's findings could contribute to advancements in AFIB detection and other ECG-related downstream tasks.

The experiments highlight the importance of selecting an appropriate context window size of 128 tokens for the downstream task to detect AFIB. More extended context window sizes do not improve performance and may introduce noise. We observed a trade-off between including more information and maintaining its relevance to AFIB detection.

Finally, this experiment's findings have implications for future research in AFIB detection and other downstream tasks for medical diagnosis based on automated interpretation of ECG. They suggest that researchers consider the optimal context window size when designing models and that shorter context windows are more effective for finetuning than previously thought. This opens the possibility of training foundation models with vast amounts of ECG data with different context window sizes, then finetuning them to a different downstream task with a much smaller portion of labeled data and producing finetuned models that outperform today's best-performing models.

Although the main hype around transformers is related to NLP tasks and human-generated text processing, they might be useful outside languages. Transformers can efficiently analyze data series and inherited patterns,

such as ECG signals. This increases the likelihood of a much broader data corpus, not just in human languages, and opens the gate for further research in other medical branches and completely different industries.

Conclusion

In this paper, we present a new approach to generating a heart language model by training a foundation model from vast amounts of data from ECG recordings and finetuning it by training with a much smaller annotated dataset. The proof of concept was realized by detecting an AFIB rhythm episode by analyzing a sequence of heartbeat annotations. We created an extensive dataset as a collection of 12 ECG benchmark databases, created an encoding scheme for transforming differences between consecutive heartbeat-to-heartbeat intervals into an alphabetical character, trained a tokenizer, developed a foundation model, and finetuned the developed model to detect AFIB on MITDB.

The achieved performance results of F1 scores of 91.97 %, 93.33 %, and 87.37 % evaluating respectively the AFDB, LTAFDB, and TDB (collection of ten ECG databases excluding the MITDB that was used for training the finetuning) achieved remarkable performance using the duration-based evaluation with the inter-patient approach to test datasets different from those trained. The principal outcome of this research is achieved using a foundation model with a context window size of 128 tokens and finetuned on the MITDB. Reviewing existing literature, we observed that limited research had been published on models demonstrating performance metrics obtained through training on one dataset and testing on an entirely different one.

AFDB was evaluated with an actual implementation of the QRS detector[60] within the ViewECG[61] monitoring and reporting solution certified as medical device software since the original AFDB does not contain beat annotations, which are assumed as input in our model. In addition, we also tested the model performance with other databases using the ViewECG QRS detector[61], achieving a slightly lower performance (less than 1%) than the performance in Table 4.

Our conclusions extracted from the results of our research are as follows. Using Transformers-based foundation models trained over a considerable amount of training data extracted from 12 ECG benchmark databases in a self-supervised way and finetuning them to specific downstream tasks using labeled training data with a much lower volume of several magnitudes (MITDB) is fruitful. We proved that the transformer-based approach produces models that can be used in real-world cases and scenarios.

Considering the effort required to generate labeled training data involving highly experienced cardiologists, this could open up the possibility of training Transformer-based models in a self-supervised manner, with unlabeled data only once, resulting in an ECG foundation model. This model can be finetuned to perform many downstream tasks requiring much lower volumes of expensive labeled training data, such as producing models that make predictions based on ECG records as input.

Given the considerable effort required to generate labeled training data, which requires the involvement of renowned trained cardiologists, we provided proof of developing an ECG foundation model through self-supervised training of Transformer-based models using unlabeled data. Applying this model to refine downstream tasks requiring significantly smaller quantities of costly labeled training data makes it possible to generate models that generate diverse predictions using ECG records as input.

However, without a compelling justification that the model with 128 tokens at the input performs best for other downstream tasks, we consider that a model with a different context window size might perform better for another downstream task is appropriate. As a result, having a range of models trained with various window sizes and testing each for different downstream classification tasks may be a feasible method and a promising area for future research.

Data availability

The databases (except AHADB) used in this research are publicly accessible on the PhysioNet[22] website[13]. They are subject to the Open Data Commons Attribution License (ODC-By) v1.0, which permits users to copy, distribute, and use the database to produce works from them, to modify, transform, and build upon the databases as long as they attribute any public use of the database, or works produced from the database, in the manner specified in the license. The license type for the AHADB is not explicitly stated. Still, it is implied to be a limited, royalty-free permission to reproduce portions of the National Uniform Billing Code (NUBC) UB-04 Data Specifications Manual and a limited license to use NUBC UB-04 Specifications Data in CMS publications.

Received: 13 June 2024; Accepted: 23 December 2024
Published online: 14 February 2025

References

1. Nesheiwat, Z., Goyal, A., Jagtap, M. & Shammas, A. Atrial fibrillation (nursing). In *StatPearls* [Internet] (StatPearls Publishing, 2022).
2. Faust, O., Ciaccio, E. J. & Acharya, U. R. A review of atrial fibrillation detection methods as a service. *Int. J. Environ. Res. Public Health* **17**, 3093 (2020).
3. Zungsontiporn, N. & Link, M. S. Newer technologies for detection of atrial fibrillation. *Bmj* **363**, k3946 (2018).
4. Rizwan, A. et al. A review on the state of the art in atrial fibrillation detection enabled by machine learning. *IEEE Rev. Biomed. Eng.* **14**, 219–239 (2020).
5. Vaswani, A. *et al.* Attention is all you need. In *Advances in Neural Information Processing Systems* 30 (2017).
6. Radford, A. et al. Language models are unsupervised multitask learners. *OpenAI blog* **1**, 9 (2019).
7. Liu, Y. *et al.* Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019).
8. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).
9. Zhuang, F. et al. A comprehensive survey on transfer learning. *Proc. IEEE* **109**, 43–76 (2020).

10. Pyysalo, S., Kanerva, J., Virtanen, A. & Ginter, F. Wikibert models: deep transfer learning for many languages. arXiv preprint arXiv:2006.01538 (2020).
11. Han, W., Pang, B. & Wu, Y. Robust transfer learning with pretrained language models through adapters. arXiv preprint arXiv:2108.02340 (2021).
12. Qasim, R., Bangyal, W. H., Alqarni, M. A. & Almazroi, A. A. A fine-tuned BERT-based transfer learning approach for text classification. *J. Healthc. Eng.* **2022**, 3498123 (2022).
13. Physionet ECG benchmark databases. https://www.physionet.org/about/database/ (2024). Accessed 27 Sep 2024.
14. Moody, G. B. & Mark, R. G. The impact of the MIT-BIH arrhythmia database. *IEEE Eng. Med. Biol. Mag.* **20**, 45–50 (2001).
15. Moody, G. A new method for detecting atrial fibrillation using rr intervals. *Proc. Comput. Cardiol.* **10**, 227–230 (1983).
16. Petrutiu, S., Sahakian, A. V. & Swiryn, S. Abrupt changes in fibrillatory wave characteristics at the termination of paroxysmal atrial fibrillation in humans. *Europace* **9**, 466–470 (2007).
17. Fuster, V. et al. ACC/AHA/ESC 2006 guidelines for the management of patients with atrial fibrillation: A report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines and the European Society of Cardiology Committee for Practice Guidelines (writing committee to revise the 2001 guidelines for the management of patients with atrial fibrillation): Developed in collaboration with the European Heart Rhythm Association and the Heart Rhythm Society. *Circulation* **114**, e257–e354 (2006).
18. Steinberg, J. S., O'Connell, H., Li, S. & Ziegler, P. D. Thirty-second gold standard definition of atrial fibrillation and its relationship with subsequent arrhythmia patterns: Analysis of a large prospective device database. *Circ. Arrhythm. Electrophysiol.* **11**, e006274 (2018).
19. American Heart Association, C. o. E., Council on Clinical Cardiology & Electrophysiology, C. American heart association ECG database (1977).
20. Nolle, F., Badura, F., Catlett, J., Bowser, R. & Sketch, M. CREI-GARD, a new concept in computerized arrhythmia monitoring systems. *Comput. Cardiol.* **13**, 515–518 (1986).
21. Taddei, A. et al. The European ST-T database: Standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography. *Eur. Heart J.* **13**, 1164–1172 (1992).
22. Goldberger, A. L. et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation* **101**, e215–e220 (2000).
23. Jager, F. et al. Long-term St database: A reference for the development and evaluation of automated ischaemia detectors and for the study of the dynamics of myocardial ischaemia. *Med. Biol. Eng. Comput.* **41**, 172–182 (2003).
24. Albrecht, P. *ST segment characterization for long term automated ECG analysis*. Ph.D. Thesis. Massachusetts Institute of Technology, Department of Electrical Engineering (1983).
25. Greenwald, S. D., Patil, R. S. & Mark, R. G. *Improved Detection and Classification of Arrhythmias in Noise-Corrupted Electrocardiograms Using Contextual Information* (IEEE, 1990).
26. IEC. IEC 60601-2-47:2012 medical electrical equipment—part 2-47: Particular requirements for the basic safety and essential performance of ambulatory electrocardiographic systems (2012). Last accessed online on 19 Jan 2024 https://webstore.ansi.org/standards/aami/ansiaamiiec60601472012r2016.
27. ANSI/AAMI. ANSI/AAMI EC57:2012 (R2020) testing and reporting performance results of cardiac rhythm and st segment measurement algorithms (2012). Last accessed online on 19 Jan 2024 at https://webstore.ansi.org/standards/aami/ansiaamiec572012r2020.
28. Lian, J., Wang, L. & Muessig, D. A simple method to detect atrial fibrillation using RR intervals. *Am. J. Cardiol.* **107**, 1494–1497 (2011).
29. Hugging Face Inc. Hugging face—the AI community building the future. https://huggingface.com (2016).
30. Gallé, M. Investigating the effectiveness of BPE: The power of shorter sequences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1375–1381 (2019).
31. Song, X., Salcianu, A., Song, Y., Dopson, D. & Zhou, D. Fast wordpiece tokenization. arXiv preprint arXiv:2012.15524 (2020).
32. Mielke, S. J. *et al.* Between words and characters: a brief history of open-vocabulary modeling and tokenization in NLP. arXiv preprint arXiv:2112.10508 (2021).
33. Benítez-Andrades, J. A., Alija-Pérez, J.-M., Vidal, M.-E., Pastor-Vargas, R. & García-Ordás, M. T. Traditional machine learning models and bidirectional encoder representations from transformer (BERT)-based automatic classification of tweets about eating disorders: Algorithm development and validation study. *JMIR Med. Inform.* **10**, e34492 (2022).
34. Matthews, B. W. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochim. Biophys. Acta (BBA)-Protein Struct.* **405**, 442–451 (1975).
35. Youden, W. J. Index for rating diagnostic tests. *Cancer* **3**, 32–35 (1950).
36. Gusev, M. & Boshkovska, M. Performance evaluation of atrial fibrillation detection. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 342–347 (IEEE, 2019).
37. Hugging Face Inc. huggingface/transformers. https://github.com/huggingface/transformers (2018).
38. Weight and Biases Inc. Weights and biases—the AI developer platform. https://wandb.ai/ (2020).
39. Pereira, T. et al. Photoplethysmography based atrial fibrillation detection: A review. *NPJ Dig. Med.* **3**, 3 (2020).
40. Lesne, A. Shannon entropy: A rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics. *Math. Struct. Comput. Sci.* **24**, e240311 (2014).
41. Afdala, A., Nuryani, N. & Nugroho, A. S. Automatic detection of atrial fibrillation using basic Shannon entropy of RR interval feature. *J. Phys. Conf. Ser.* **795**, 012038 (2017).
42. Canova, F. Detrending and turning points. *Eur. Econ. Rev.* **38**, 614–623 (1994).
43. Dash, S., Chon, K., Lu, S. & Raeder, E. Automatic real time detection of atrial fibrillation. *Ann. Biomed. Eng.* **37**, 1701–1709 (2009).
44. Lee, J., Nam, Y., McManus, D. D. & Chon, K. H. Time-varying coherence function for atrial fibrillation detection. *IEEE Trans. Biomed. Eng.* **60**, 2783–2793 (2013).
45. Zhao, H., Lu, S., Zou, R., Ju, K. & Chon, K. H. Estimation of time-varying coherence function using time-varying transfer functions. *Ann. Biomed. Eng.* **33**, 1582–1594 (2005).
46. Tateno, K. & Glass, L. A method for detection of atrial fibrillation using RR intervals. In *Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163)*, 391–394 (IEEE, 2000).
47. Zhou, Y. *et al.* Masked transformer for electrocardiogram classification. arXiv preprint arXiv:2309.07136 (2023).
48. Zhao, Z. Transforming ECG diagnosis: An in-depth review of transformer-based deeplearning models in cardiovascular disease detection. arXiv preprint arXiv:2306.01249 (2023).
49. Atiea, M. A. & Adel, M. Transformer-based neural network for electrocardiogram classification. *Int. J. Adv. Comput. Sci. Appl.* **13**, 11 (2022).
50. Rubel, P. *et al.* Scp-ecg v3. 0: An enhanced standard communication protocol for computer-assisted electrocardiography. In *2016 Computing in Cardiology Conference (CinC)*, 309–312 (IEEE, 2016).
51. Wagner, P. et al. Ptb-xl, a large publicly available electrocardiography dataset. *Sci. Data* **7**, 154 (2020).
52. Che, C., Zhang, P., Zhu, M., Qu, Y. & Jin, B. Constrained transformer network for ECG signal processing and arrhythmia classification. *BMC Med. Inform. Decis. Mak.* **21**, 1–13 (2021).

53. Liu, F. et al. An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. *J. Med. Imaging Health Inform.* **8**, 1368–1373 (2018).
54. Sanh, V., Debut, L., Chaumond, J. & Wolf, T. Distilbert, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019).
55. Varghese, A., Kamal, S. & Kurian, J. Transformer-based temporal sequence learners for arrhythmia classification. *Med. Biol. Eng. Comput.* **61**, 1993–2000 (2023).
56. Nozza, D., Bianchi, F. & Hovy, D. What the [mask]? making sense of language-specific BERT models. arXiv preprint arXiv:2003.02912 (2020).
57. Dong, Y., Zhang, M., Qiu, L., Wang, L. & Yu, Y. An arrhythmia classification model based on vision transformer with deformable attention. *Micromachines* **14**, 1155 (2023).
58. Pereira, R. & Andreão, R. V. Inter-patient detection of atrial fibrillation in short ECG segments based on LSTM network with multiple input layers. *Res. Biomed. Eng.* **38**, 465–476 (2022).
59. Jahan, M. S., Mansourvar, M., Puthusserypady, S., Wiil, U. K. & Peimankar, A. Short-term atrial fibrillation detection using electrocardiograms: A comparison of machine learning approaches. *Int. J. Med. Inform.* **163**, 104790 (2022).
60. Domazet, E. & Gusev, M. Improving the QRS detection for one-channel ECG sensor. *Technol. Health Care* **27**, 623–642 (2019).
61. ViewECG: A platform for ECG monitoring and reporting tools. https://viewecg.com/ (2020). Accessed on 10 Sep 2024.

Acknowledgements
This research was supported by the project entitled "AI Cardiologist - Alerting on Dangerous Arrhythmia" within the ELISE Open Call, funded by the EU H2020 project no.951847.

Author contributions
S.T. contributed to the initial idea, conducted the experiments, M.G. conceived the heart-based language encoding and feature engineering, analyzed data and results, provided domain-specific explainability, E.K. supervised the methods and results, All authors reviewed the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to M.G.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.