Towards Trustworthy Atrial Fibrillation Classification From Wearables Data: Quantifying Model Uncertainty

Ciaran Bench[1], Nils Strodthoff[2], Mohammad Moulaeifard[2], Philip Aston[1,3], Andrew Thompson[1]

[1] National Physical Laboratory, Teddington, United Kingdom
[2] Department of Health Services Research, Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany
[3] Department of Mathematics, University of Surrey, Guildford, United Kingdom

Abstract

Wearable devices capable of acquiring photoplethysmography (PPG) signals are increasingly used to monitor patient health outside of typical clinical settings. PPG signals encode information about relative changes in blood volume and, in principle, can be used to assess various aspects of cardiac health non-invasively, e.g. to detect atrial fibrillation (AF). Machine learning based techniques have clear potential to automate diagnostic protocols for AF, where deep networks have been shown particularly effective. However, these models are prone to learning biases and lack interpretability, leaving considerable risk for poor generalisability and misdiagnosis. In order to make these models suitable for routine use in clinical workflows, the uncertainty of a model's output should be quantified to establish whether it can reliably inform diagnoses. Here, we describe the use of Monte Carlo Dropout to estimate the uncertainties of deep learning models trained to predict AF from PPG time series.

1. Introduction

Changes in blood flow and composition can be used to monitor several indicators of patient health, such as cardiac rhythm, arterial blood oxygen saturation, blood pressure, atrial fibrillation, and heart rate [1]. This information can be detected from photoplethysmography (PPG) signals, that encode changes in the optical properties of the skin's microvascular bed driven by variations in blood volume.

Recent innovation in sensor manufacturing has given rise to wearable PPG sensing technologies that have clear potential to facilitate patient monitoring outside of typical clinical settings. Such capabilities could drastically improve the ease, frequency, and accuracy with which one can detect risk factors related to cardiovascular disease [2]. This is particularly the case for the detection of atrial fibrillation (AF), a condition characterised by short periods of rapid, uncoordinated heart beat that correspond with an increased risk of stroke and other ailments. Anticoagulation treatments are effective in reducing this risk, but episodes are underdiagnosed using current clinical procedures [3]. The use of wearables data could enhance screening capabilities by improving the ease of measurement, providing a means for continuous monitoring, and by increasing the availability of measurement devices. Therefore, their use holds considerable potential to dramatically improve patient outcomes.

However, it is not feasible for human clinicians to manually analyse the large quantities of time-series data collected with these devices. As a result, the realisation of these new sensing capabilities hinges on the development of automated techniques for analysing PPG data.

Statistical/analytical prediction models use features manually-extracted from the raw time series as inputs. For example, those parsed from Poincaré plot analysis, successive differences in peak to peak interval lengths, signal slope changes, and Shannon entropy [4]. In contrast, deep networks can automatically learn relevant features from the raw time series, offering the potential to provide significant performance benefits. This is particularly the case where feature extraction is not robust to noise/artefacts (deep learning models implicitly take this into account during training), or where prior knowledge of the most relevant features is incomplete or not completely accurate. This approach has been used for AF classification, achieving good performance [5].

However, where normal heart rhythm is indicated by regular intervals between pulses with little variance in their morphology, AF, on the other hand, may be recognised by irregular spacing between pulses and significant variance in their morphology. Consequently, the adverse effect motion can have on the morphology of PPG waveforms may be easily misinterpreted as being caused by an arrhythmia. This, in addition to the tendency for models to over-

fit, leaves considerable risk for poor generalisability and misdiagnosis [6]. Some estimate of the uncertainty of a prediction (i.e. the degree of doubt) is needed to establish whether it can be used to reliably inform diagnosis.

There are several approaches for estimating the uncertainty of predictions output by deep networks. Here we opt for Monte Carlo Dropout (MCD) (an approximation scheme for performing variational inference on Bayesian Neural Networks) given the relative ease with which it can be implemented compared to other approaches, as well as its efficiency, making its use tractable on larger models trained on larger datasets [7]. With this technique, a model is trained with dropout applied liberally throughout the architecture. During evaluation, dropout is left active, and a given input is evaluated several times, resulting in a distribution of predictions for a given input (i.e. in contrast with the more typical case where a network takes a single input and produces a single prediction – with this technique, we get several different predictions for a given input). The variance in these predictions encodes predictive uncertainty, where the procedure used to express uncertainties depends on the training task. With that said, the magnitudes of the estimated uncertainties are known to depend heavily on the values of certain hyperparameters (in particular the dropout rate) [8]. Therefore, a grid search is advised to derive hyperparameter values that result in well-calibrated uncertainties (i.e. large uncertainties correspond with incorrect predictions and *vice versa*).

2. Methods

There are several sources of uncertainty that may be considered, though the two most significant sources are typically epistemic uncertainty (model uncertainty, that may be reduced with more data or a more suitable architecture), and aleatoric uncertainty (irreducible data uncertainty, e.g. noise). Here we model both by learning the optimal parameters θ of a deep neural network model G_θ, which is an xresnet1d-50 [9] model, i.e., a one-dimensional adaptation of a ResNet-style convolutional neural network often used in computer vision. Here, we apply dropout liberally throughout the architecture and add a parallel output layer composed of an adaptive pooling layer, followed by a feed-forward layer (two output nodes, each providing the predicted mean and variance of each logit) and a Softplus activation. The model is trained to output the means and variances μ_{AF}, μ_{noAF}, σ^2_{AF} and σ^2_{noAF} used to parameterise two Gaussian distributions $\mathcal{N}_{\text{AF}}(\mu_{\text{AF}}, \sigma^2_{\text{AF}})$ and $\mathcal{N}_{\text{noAF}}(\mu_{\text{noAF}}, \sigma^2_{\text{noAF}})$ that describe the logit distribution for each class. These distributions are independently sampled (using the procedure described in [7] and Algorithm 1, represented here by $\mathbf{l}_{k,t} \sim \mathcal{N}_{G_\theta}(x)$), to produce a vector of logits $\mathbf{l}_{k,t}$ where the subsequent application of a Softmax should produce a vector of probabilities such that $\text{argmax}(\mathbb{E}\{\mathbb{E}[\text{Softmax}(\mathbf{l}_{k,t} \sim \mathcal{N}_{G_\theta}(x))]_t\}_k) = \text{argmax}(y_i)$. Here, $y_1, y_2, ...y_i, ...y_N \in Y$ are the corresponding one-hot encoded classification labels, and where the model is given a PPG time series $x_1, x_2, ...x_i, ...x_N \in X$ as an input. K is the number of Monte Carlo samples (i.e. number of times each input example is evaluated, with each evaluation indexed with k), and T is the number of noise samples (where each sample is indexed with t).

We train our model on the DeepBeat dataset [5], with a custom split of the dataset to ensure no patient overlap and a balanced label distribution across training, validation and test set (see Table 1).

Table 1. Modified DeepBeat dataset

Set	Patients	Total samples	AF samples	Non-AF samples
Train	85	106249	40603	65646
Validation	25	15256	5800	9456
Test	24	15377	5797	9580

The loss and training procedure is the same as that described in [7]. The minimum of the validation loss was used as the stopping criteria. Evaluation was performed according to Algorithm 1, where the operator H computes the entropy.

Algorithm 1 Monte Carlo Dropout Classification Evaluation Loop

for each input **do**
  for $k = 1$ to K **do**
    Acquire the predicted mean $f_{i,k}$ and variance $\sigma^2_{i,k}$ for each logit i
    for $t = 1$ to T **do**
      for each logit i **do**
        Sample $\epsilon_{i,k,t} \sim N(0, 1)$
        Compute sampled logit: $l_{i,k,t} = f_{i,k} + \sigma^2_{i,k}\epsilon_{i,k,t}$
      end for
      Compute $\mathbf{p}_{k,t} = \text{Softmax}(\mathbf{l}_{k,t})$
    end for
    Compute average $\bar{\mathbf{p}}_k = \frac{1}{T}\sum_{t=1}^{T} \mathbf{p}_{k,t}$
  end for
  $H_{\text{Total}} = H(\frac{1}{K}\sum_{k=1}^{K} \bar{\mathbf{p}}_k)$
end for

The total uncertainty (combination of aleatoric and epistemic) is given by H_{Total} in Algorithm 1. We use the uncertainty calibration error (UCE) proposed in [10] to validate the uncertainties predicted by our model. Here, the estimated uncertainty is binned into equal-length intervals, and the average number of incorrect responses in each bin as well as the average entropy in each bin is calculated. The UCE is designed to assess the linearity of the correlation between the average entropy and the average in-

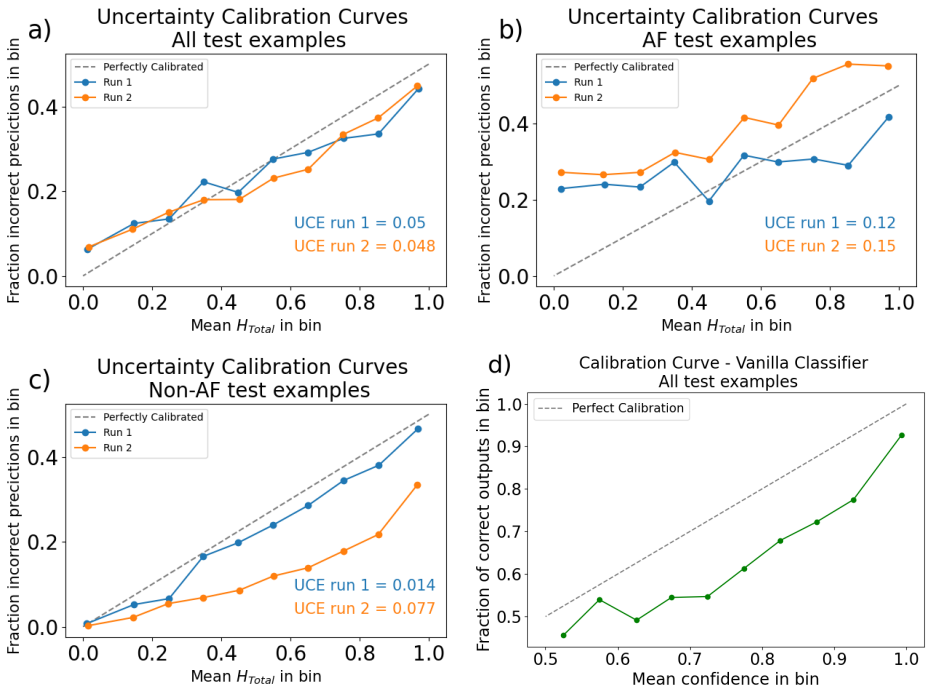

Figure 1. Uncertainty calibration curves for a) all test examples b) AF test examples, and c) Non-AF test examples produced from two training/evaluation runs of a model trained and evaluated with a 3 % dropout rate. While a repeat training run may result in small variations in predictive performance and the calibration when assessed over the whole test set, the calibration of a given class can vary considerably. d) shows the calibration curve for the vanilla classifier.

accuracy of the binned estimates, where a slope of 1 is considered a perfect calibration [10]. In the binary classification case, the maximum entropy (1 when expressed in bits) should correspond to an accuracy of 50 %, so, here, a slope of 0.5 corresponds to a perfect calibration. The UCE is expressed as,

$$\mathrm{UCE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left| \mathrm{err}(B_m) - \frac{\mathrm{uncert}(B_m)}{2} \right|, \quad (1)$$

where B_m are the predictions in a bin (indexed by m) parsed from all n test examples,

$$\mathrm{err}(B_m) := \frac{1}{|B_m|} \sum_{i \in B_m} 1(\hat{y}_i \neq y_i) \quad (2)$$

and,

$$\mathrm{uncert}(B_m) := \frac{1}{|B_m|} \sum_{i \in B_m} H_{\mathrm{Total},i}, \quad (3)$$

where $\hat{y}_i$ is the one-hot encoded predicted class. We consider a single dropout rate (3 %), and compute the UCE and plot uncertainty calibration curves for the test sets for two repeat training/evaluation processes, each time using a different random weight initialisation. We also assess calibration quality conditioned on each class by plotting uncertainty calibration curves for the subset of examples known to exhibit an AF, and for those that do not. We also train a model without implementing Monte Carlo Dropout (using an architecture with only a single output branch, trained with a generic cross-entropy loss) to illustrate how the output probabilities themselves can be interpreted as uncertainties. We compute the expected calibration error (ECE) to assess calibration quality for this model,

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left| \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \right|, \quad (4)$$

where the accuracy $\mathrm{acc}(B_m)$ is given by

$$\mathrm{acc}(B_m) := \frac{1}{|B_m|} \sum_{i \in B_m} 1(\hat{y}_i = y_i) \quad (5)$$

and the confidence $\mathrm{conf}(B_m)$ is given by

$$\mathrm{conf}(B_m) := \frac{1}{|B_m|} \sum_{i \in B_m} p_i, \quad (6)$$

where p_i is the probability associated with the predicted class of each example in a bin [11].

3. Results

Summary metrics over the whole test set are shown in Table 2 for each training run (referred to as 'Run 1' and

'Run 2)', indicating the model trained with Monte Carlo Dropout has comparable performance with the existing literature [5]. While a repeated training run produced small differences in overall predictive performance and the UCE when assessed over the whole test set (see Fig. 1a), significant differences were found in calibration quality when assessed individually for each class (see Figs. 1b and c). These results emphasize that validation metrics computed over the whole test set may not be adequate for understanding all the properties that could impact the practical utility of the model in clinical scenarios (poor calibration on the positive class) or the repeatability of a model's calibration. Table 2 and Fig. 1d show the results of the vanilla classifier trained without Monte Carlo Dropout. The implementation of Monte Carlo Dropout does not appear to degrade performance, producing similar AUC and F1 values. The use of class probabilities as a proxy for confidence also appears to produce calibrated uncertainties (albeit, less well calibrated than those produced with Monte Carlo Dropout). However, given that the uncertainties are not modelled explicitly with this approach, this framework does not enable further decomposition of the various sources of uncertainty that might otherwise be used to assist with model prototyping or dataset curation.

Table 2. AF Classification: Test Set

Run	Accuracy	AUC	F1	Calibration
MCD 1	0.79	0.87	0.71	UCE = 0.050
MCD 2	0.78	0.85	0.67	UCE = 0.048
Vanilla	0.79	0.87	0.72	ECE = 0.095

4. Conclusion

Here, we have implemented Monte Carlo Dropout on an AF classifier trained on a custom split of the DeepBeat dataset. While a cursory glance at a calibration curve (e.g. Fig. 1 a) might suggest the model is producing well-calibrated predictions, there can be significant variation across classes and repeat runs. Caution should be exercised when using estimated uncertainties for diagnostic purposes. A careful evaluation of calibration quality is needed to devise suitable strategies for using predictions to inform diagnoses. For example, here we have shown that conditional calibration metrics (per-class UCE) can be used to acquire a more comprehensive assessment of performance.

Furthermore, given that the quality and composition of the estimated uncertainties is model/task dependent, we recommend experimenting with a range of architectures and hyperparameters. While not performed here, expressing the estimated uncertainties in a way that disentangles the aleatoric uncertainty could be used to aid in model prototyping/dataset curation (e.g. to determine whether more data of a given class is needed to reduce any potential variation in uncertainty composition across classes).

Acknowledgments

The project (22HLT01 QUMPHY) has received funding from the European Partnership on Metrology, co-financed from the European Union's Horizon Europe Research and Innovation Programme and by the Participating States. Funding for NPL was provided by Innovate UK under the Horizon Europe Guarantee Extension, grant number 10084125.

References

[1] Park J, Seok HS, Kim SS, Shin H. Photoplethysmogram Analysis and Applications: an Integrative Review. Frontiers in Physiology 2022;12:808451.

[2] Charlton PH, Kyriacou PA, Mant J, Marozas V, Chowienczyk P, Alastruey J. Wearable Photoplethysmography for Cardiovascular Monitoring. Proceedings of the IEEE 2022;110(3):355–381.

[3] Zungsontiporn N, Link MS. Newer Technologies for Detection of Atrial Fibrillation. BMJ 2018;363.

[4] Pereira T, Tran N, Gadhoumi K, Pelter MM, Do DH, Lee RJ, Colorado R, Meisel K, Hu X. Photoplethysmography Based Atrial Fibrillation Detection: A Review. NPJ Digital Medicine 2020;3(1):1–12.

[5] Torres-Soto J, Ashley EA. Multi-task Deep Learning for Cardiac Rhythm Detection in Wearable Devices. NPJ Digital Medicine September 2020;3(1):116. ISSN 2398-6352.

[6] Nie G, Zhu J, Tang G, Zhang D, Geng S, Zhao Q, Hong S. A Review of Deep Learning Methods for Photoplethysmography Data. arXiv preprint arXiv240112783 2024;.

[7] Kendall A, Gal Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? Advances in Neural Information Processing Systems 2017;30.

[8] Gal Y, Hron J, Kendall A. Concrete Dropout. Advances in Neural Information Processing Systems 2017;30.

[9] Strodthoff N, Wagner P, Schaeffter T, Samek W. Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL. IEEE Journal of Biomedical and Health Informatics 2020;25(5):1519–1528.

[10] Laves MH, Ihler S, Kortmann KP, Ortmaier T. Calibration of Model Uncertainty for Dropout Variational Inference. arXiv preprint arXiv200611584 2020;.

[11] Guo C, Pleiss G, Sun Y, Weinberger KQ. On Calibration of Modern Neural Networks. In International Conference on Machine Learning. PMLR, 2017; 1321–1330.

Address for correspondence:

Andrew Thompson

National Physical Laboratory, Hampton Road, Teddington, Middlesex, TW11 0LW, United Kingdom

andrew.thompson@npl.co.uk