Quantifying Uncertainty in Blood Oxygen Estimation Models from Real-World Data

Gianluca Truda[2670558], Serafim Korovin[2666465], and Adam Kantorik[2651160]

Group 123
Data Mining Techniques
Vrije Universiteit Amsterdam

1 Introduction

1.1 Background

The blood oxygen level of a patient is an important clinical metric that is useful in the diagnosis and monitoring of respiratory illnesses, including *Covid-19*[2]. Arterial oxygen saturation can be estimated through non-invasive methods with high levels of accuracy, such as by measuring the perfusion of blood to the skin. This peripheral oxygen saturation (SpO2) level is measured with a pulse oximeter device that leverages the different absorbance levels of oxy- and deoxy-haemoglobin at 600nm and 940nm wavelengths [10]. A photoplethysmogram (PPG) signal is constructed from the raw input of the photodiode. This typically involves filtering frequencies outside of a specific range and correcting for the light source used. Unfortunately, this specialised hardware can be cost-prohibitive and under-supplied in times of crisis, such as the *Covid-19* pandemic.

1.2 Related Work

Recently, some studies [10,12,9,15,13,14] have investigated if smartphone cameras could be used to estimate SpO2 levels from direct contact with a patient's finger. This has the potential to make pulse oximetry widely available. Notably, Lamonaca et al. [10] used a knowledge-based approach to achieve similar accuracy to a commercial pulse oximeter using only videos captured of patients' fingers with a smartphone. These results were replicated in a study by Nemcova et al. [12], but required arbitrary adjustments to the Lamonaca formula. Most of the studies relied on using a specific smartphone and tuned the model to its hardware properties.

1.3 Aims

The open source *CoVital* project [6] aims to recreate the accuracy of commercial pulse oximeters on any modern smartphone. The team of volunteers has worked to compile a dataset from multiple sources – the Nemcova study, volunteers with pulse oximeters, and clinicians treating *Covid-19* patients – and is

employing machine learning to produce accurate SpO2 estimation models that can be deployed in a smartphone app.

Training models for medical tasks introduces a host of unique challenges [3]. Notably, it is valuable to have not only predictions, but estimates of the confidence of those predictions – which help clinicians and patients make better decisions about when to trust the model. Without such confidence estimates, even the most accurate models in evaluation may be dangerous in deployment [11]. To address this issue in the context of smartphone-driven SpO2 estimation, we evaluated the use of dropout techniques in a hybrid deep learning model to generate 95% confidence intervals (CIs).

2 Materials and Methods

2.1 Dataset

Four sources of data were used to construct the dataset for this study:

1. **Nemcova data**: The publicly-accessible videos from the study by Nemcova et al. [12]: 49 samples.
2. **Sample data**: Videos and ground truth readings from an early *CoVital* volunteer with access to a pulse oximeter: 11 samples.
3. **Community data**: Videos and readings from healthy volunteers with access to pulse oximeters, collected through the *CoVital* application: 19 samples
4. **Clinical data**: Videos and readings from healthy and unhealthy patients collected by medical doctors partnered with the *CoVital* project: 6 samples.

Transforming Videos to Time Series The datasets consisted of 30 fps HD videos of the patient's finger and JSON files with ground-truth measurements (including SpO2 and heart rate). A decoder based on *openCV* was used to read the video files and convert each frame to a vector of floating-point values corresponding to the mean and standard deviation for each of the three colour channels; with integer indexes for the frame number and sample ID, and a reference to the video source file. This was repeated over every frame to produce an $N \times 9$ DataFrame for each sample (see Table 1). These were stitched together vertically.

Table 1. An illustrative sample of time series data after conversion from mp4 video files. Floating point values and filepaths are truncated for formatting purposes.

frame	sample	mean_red	std_red	mean_green	std_green	mean_blue	std_blue	source
0	0	231.99	0.1425	58.23	2.6861	43.84	3.5604	[filepath]
1	0	231.98	0.1576	54.83	2.6311	43.66	3.5964	[filepath]
2	0	231.98	0.1832	55.33	2.8542	43.59	3.6227	[filepath]

The ground truth data from the JSON files was extracted to a single $N \times 3$ DataFrame, which could be joined with the time series data based on the source video's file path (see Table 2).

Table 2. An illustrative sample of the ground truth data paired with each video. Floating point values are truncated for formatting purposes. HR = hear rate (in beats per minute), SpO2 = peripheral blood oxygen saturation (%).

path	**HR**	**SpO2**
sample_data/20200327153700000000	67	98
sample_data/20200327151800000000	66	98
sample_data/20200329132000000000	60	97

Unfortunately, after transforming the data to time series, it was discovered that 10 of the 11 videos from the **sample data** collection had little-to-no data in the green channel. This may have been due to some automatic colour correction by the smartphone camera app. Because of this anomaly, the 11 videos from that data source were excluded.

The remaining 74 videos from the **Nemcova**, **community**, and **clinical** sources were considered for the final dataset. A further 7 were excluded for having missing ground-truth values for SpO2, or other anomalies that rendered the data unusable. The remaining 67 samples were combined into a single dataset.

Analysis of Distributions There was a mean of 583.4 frames per sample, with a minimum of 290 frames and a maximum of 900 frames. Inspection of the distribution (see Fig. 1) revealed a bimodal pattern, which was due to the difference in methodologies between the Nemcova et al. study [12] and the *CoVital* data capture application. The former captured around 10 seconds per sample, whilst the latter captured around 30 seconds per sample. Because only a few seconds of data should, in principle, be required for SpO2 estimation [10,12], this opened up the possibility of data augmentation (see section 2.2).

The pixel summary statistics exhibited skewed distributions[1] for both the mean and standard deviation (see Fig. 2). The means of blue and green were positively skewed (2.84 and 0.92) over the whole dataset, whilst the mean red was negatively skewed (-1.18). The standard deviations for each colour channel were all positively skewed (0.26, 0.74, 3.19).

Analysis of Correlations The relationships between variables were analysed using Pearson's correlation coefficient. The ground truth values of heart rate (HR) and peripheral oxygen saturation (SpO2) had a negligible negative correlation, which tracked with the priors from physiology. There were no notable correlations between the frame number or sample ID and any of the colour parameters, but there were some strong positive and negative correlations between

[1] measured with Pearson's moment coefficient of skewness

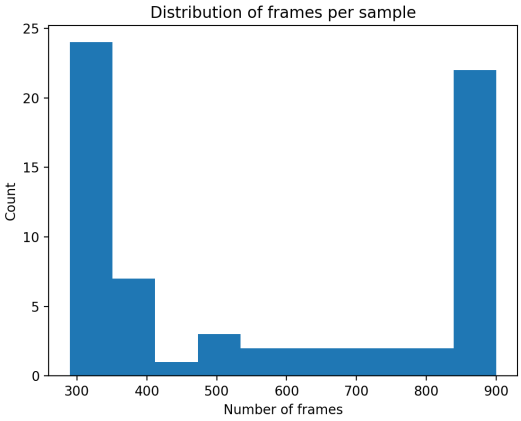

Fig. 1. Distribution of the length of the original 67 samples (in frames), illustrating why data augmentation was both necessary and feasible.

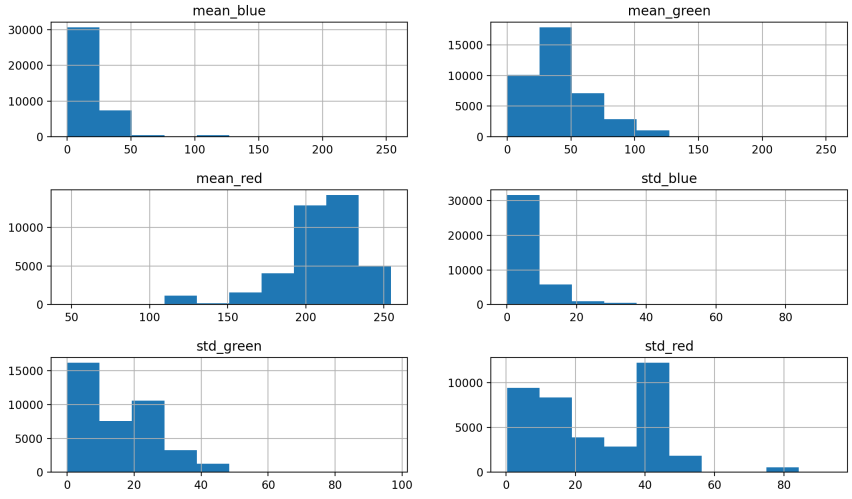

Fig. 2. Distributions of the raw time series values across all samples after decoding the mp4 videos.

the colour parameters themselves (see Fig. 3). This can be explained by the inherent relationship between the mean and standard deviation of a distribution, as well as the fact that changes in white light levels during measurement would have affected all three colour channels similarly.

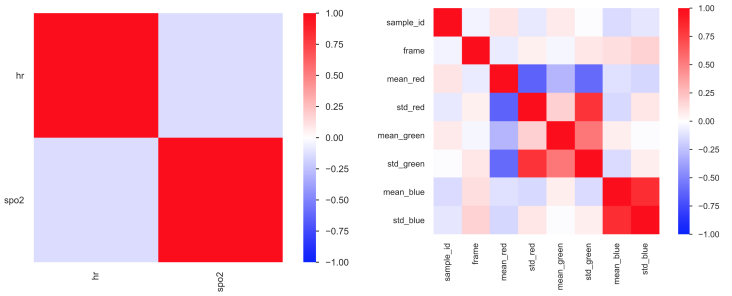

Fig. 3. Pearson correlation matrices for ground truths (left) and time series data (right).

Target Features Unfortunately, the ground truth values were not evenly distributed (see Fig. 4). SpO2 was strongly negatively skewed (-2.82), with over 76% of the values being in the range 97-99. Only 3 samples out of the 67 in our combined dataset had values sufficiently low to be considered a health risk [2]. This was due to the fact that most of the data came from healthy volunteers with SpO2 values in the typical 97-99 range. Over time, it is likely that more data will be provided by clinicians treating patients with low SpO2 values, but that fell outside the scope of this study. The medical consultants on the *CoVital* project defined an acceptable error of approximately 1% in either direction, but 73% of the SpO2 samples fell within this distance from the mean of 97.1. Unfortunately, this made it very difficult to evaluate generalisable model performance. That is to say, a naive model that always predicted an SpO2 of 97 would have been within the acceptable error bounds nearly $\frac{3}{4}$ of the time. Clearly, such a model would not only be unhelpful but, indeed, dangerous.

Luckily, the heart rate (HR) values of the samples were much more evenly distributed (see Fig. 4), with a skewness of only 0.19. We thus opted to use HR as a proxy target feature. This allowed us to validate our methodology on a very similar biological attribute, using identical input data and the same data pipelines. Because the underlying data and the methodology were identical, we were confident that our results on HR estimation could be generalised to SpO2 estimation[2].

[2] subject to the assumption that models could later be trained on more evenly-distributed SpO2 values as new data was captured.

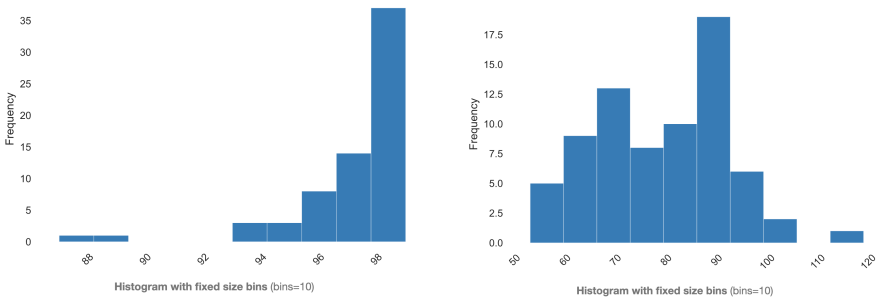

Fig. 4. Distribution of peripheral oxygen saturation (left) and heart rate (right) ground truths across all 67 initial samples.

2.2 Preprocessing

Sample Augmentation Samples from the *CoVital* volunteers and partner clinicians comprised a large portion of the combined dataset, but most samples were between 800 and 900 frames in length. Conversely, the data sourced from Nemcova et al. [12] was typically of 300-400 frames per sample. Commercial pulse oximeters only require a few seconds of clean signal to make an accurate estimate – corresponding to a few hundred frames. This afforded us the opportunity to augment the dataset and enhance both the training and evaluation of our model.

Initially, we did this by simply cutting up samples into blocks of N contiguous frames. The ground truth HR value for each original sample was duplicated across each of its augmented sub-samples. We found that this improved prediction performance, so we extended the technique by developing a custom rolling function. The method trims some frames off the beginning and end of each sample, as those regions are often anomalous due to users starting and ending the recording. Next, the method copies the first N frames, shifts the index by S frames, then makes another copy of N frames in length. This is repeated until there are no longer N frames remaining to be copied. The result is a collection of overlapping sub-samples taken from the original sample. The corresponding sample ID is mutated appropriately, such that each augmented sub-sample is treated independently.

We tested a range of values for N using a grid search on a collection of generalised linear models predicting the HR labels (evaluated with mean squared error). We found robust results for $N = 200$, $S = 25$, and trimming 50 frames off the beginning and the end of each sample.

This augmentation technique increased the number of samples from 67 to 779, allowing for more extensive training and evaluation of our models. Moreover, because subsequent preprocessing and feature engineering was performed at the level of the augmented sub-samples, the degree of similarity between overlapping sub-samples was minimised to negligible levels.

Generating PPG Signal The pixel means constituted the equivalent of raw sensor input for each of the three colour channels. These were transformed into curves that approximated a photoplethysmogram (PPG) signal. This was done by applying the following preprocessing techniques to each colour channel independently:

- Applied Butterworth band-pass filter between 0.3 and 4.2 Hz, to filter out periodic signals not associated with human cardiac activity [10].
- Clipped values outside of 2.5 standard deviations from the other values in the sample. This capped the magnitude of anomalous values and prevented them from swamping others during scaling.
- Applied min-max scaler to signal, so values were between 0 and 1.0.
- Applied smoothing function (based on 2-frame rolling mean). This had the effect of a smoothing anomalous perturbations out of the signal.

The above process was applied independently to each colour channel of each (augmented) sub-sample. An example of the effects can be seen in Fig. 5.

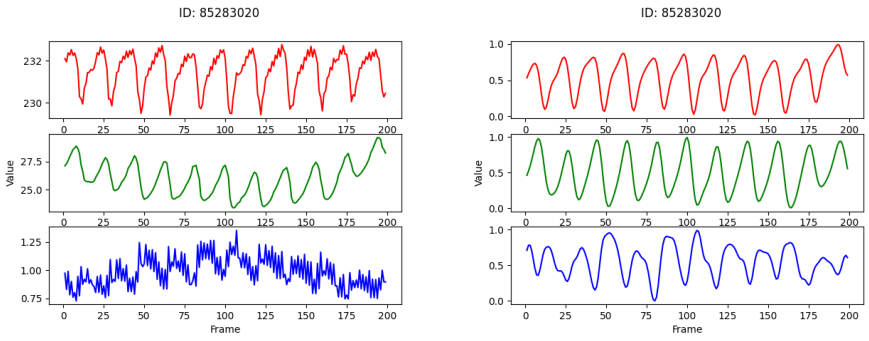

Fig. 5. Illustrative example of the smoothing and scaling effects of applying the PPG processing pipeline to raw colour channels.The left figure depicts the pixel means for each colour channel for an augmented sub-sample of 200 frames in length. The right figure is the result of applying the preprocessing tools (described above) to that sample.

2.3 Evaluating Signal Quality

Unlike purpose-built pulse oximetry hardware, smartphone cameras vary in signal quality [9] due to differences in cameras and lenses, as well as the imperfect user adherence to instructions. Therefore, an important consideration for this project was evaluating the quality of the signal. We chose to do this after the preprocessing stage, as much of the variation in the signals was removed by the time we had generated PPG waveforms.

Total Skewness Metric Previous studies on PPG signal quality compared several signal quality indices and found *skewness* to be the best performer [4]. We implemented this metric using the Fisher-Pearson coefficient of skewness, which measures where the "weight" of the distribution is for some sample of data. This was applied to each colour channel independently. Because signals can be positively or negatively skewed, we opted to combine the skewness for each channel by summing the absolute values of skewness for each colour, with scores closer to zero indicating higher quality.

Mutual Frequency Divergence Metric To complement total skewness, we developed a second metric based on the mutual Kullback-Leibler (KL) divergence across channels in the frequency domain. Our goal was to measure how similar the periodic structures of each colour channel were at frequencies corresponding to cardiac activity. We applied a Fourier transform using Welch's method [16]. This gave us the power spectrum of each channel. We selected the subset that fell between 0.8 and 2.5 Hz (where cardiac activity occurs) and then calculated the KL divergence between each pair of channels in both directions. We summed these 6 values to produce a total divergence score.

By combining these two measures of signal quality, we were able to assess the signal quality of the PPG and discard signals where anomalies had occurred during capturing of the data – such as the subject removing their finger or bumping their phone. We also fed the quality metrics into our models as features to assist in uncertainty estimation.

2.4 Features

After preprocessing, we had 6 time series vectors, namely: pixel standard deviations and transformed pixel means (PPG) for each colour channel. Some learning algorithms (such as LSTMs) are designed for time series data, whilst others require atemporal records and treat all observations as independent. To accommodate this, we needed a flexible feature engineering and selection approach. Moreover, because previous work in this domain made use of knowledge-based approaches, we had no good priors for what feature engineering techniques would work best for this task. These constraints informed our design decisions.

Feature Engineering To collapse time series data into a wide atemporal format, we utilised the *tsfresh* library to generate thousands of possible features from each of our 6 time series vectors. This was done at the level of each augmented sub-sample in order to enforce independence. This approach utilised over 50 different feature calculators, including autocorrelation, Benford correlation, Fourier coefficients, sample entropy, and least-squares linear trend. The full list is detailed in the tsfresh documentation.

Feature Selection The exhaustive feature engineering produced wide matrices that were unsuitable for modelling. We employed feature selection to extract only

the best K features based on the ANOVA F-value between each feature and the target feature (HR). As before, we made use of grid search over a collection of generalised linear models to select the optimal value for K. For $K = 50$, we found the best compromise between model bias and variance when performing 5-fold cross validation. We therefore used the 50-best features for the LSTM models.

2.5 LSTM Modelling

Model Structure We chose a deep hybrid neural network approach for our regression task. The final model architecture can be seen in Fig. 6.

Recurrent neural networks (RNNs) are the most appropriate choice when dealing with temporal data, due to their purpose-built structure. However, they are especially vulnerable to the issues of exploding or vanishing gradients. Long-Short Term Memory (LSTM) networks overcome these issues, as well as learn long-term dependencies [8]. These reasons, combined with the accessibility of the frameworks, made LSTMs a clear choice for processing the time series data. To improve our feature space, we supplemented the LSTM with a collection of 50 engineered features (as described above).

Our final model consists of two input layers. One feeds time series (PPG) data to stacked LSTM layers. The other feeds the 50-best engineered features to a dense layer providing atemporal information. The outputs of both layers are combined in a concatenation layer, after which dropouts can be applied. Finally, the connected data is directed into the last dense layer which creates the output of one value, representing a single prediction.

The feature of our model which was central to reaching our goal of estimating uncertainty was Monte-Carlo (MC) dropout. In normal dropout, the model randomly assigns zero to a specified proportion of units at training time only. On the other hand, MC dropout applies it at *both* training and inference stages. This way, we are able to obtain different predictions from the same inputs, producing prediction distributions – which can be interpreted to represent uncertainty. Additionally, signal quality is estimated for each sample using the two bespoke signal quality functions. In the dynamic version of our LSTM model, the number of dropout-driven samples is varied depending on the signal quality, thus incorporating that information into the established confidence intervals.

Hyperparameters

1. Learning rate = 0.01
2. Number of epochs = 100 (for a quicker comparison between models' performance), final model was trained on 1000 epochs
3. Batch size = 64
4. Loss function = mean squared error
5. Activation functions:
   - hyperbolic tangent for LSTM outputs (to keep the values from exploding)
   - ReLU for others (since our targets are non-zero values)
6. Dropout rate = 0.2

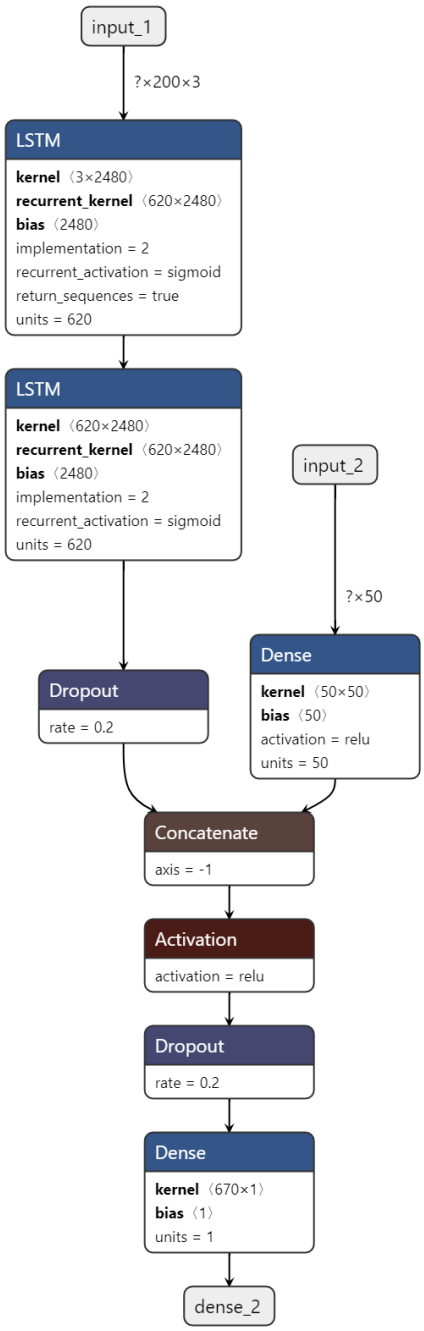

Fig. 6. Structure and key hyperparameters of our final **LSTM** model, which achieved the best overall performance in evaluation.

2.6 Random Forest Modelling

As a comparison for our hybrid LSTM model, we developed a bootstrap aggregation (bagging) ensemble, implemented with random forest regression. This was chosen for two reasons. Firstly, bagging is known for seldom overfitting to the data [1], which is an immensely valuable property on small (augmented) datasets like the one used in this study. Secondly, because the random forest approach trains each model independently in a parallel manner, it was trivial to generate confidence intervals (CIs) by simply analysing the distribution of each individual tree's predictions. Boosting would not have been suited to this task, because the estimators are trained sequentially and are dependent on one another.

As input, the ensemble used the features engineered by *tsfresh*, using the best K features selected, as well as two features generated by running both bespoke signal quality functions on the samples. To find optimal hyperparameters without overfitting, grid search with 5-fold cross-validation was used. N (the number of trees in the ensemble) converged at an optimum near 1000 trees, but K (the number of features) seemed to only increase performance (measured in mean absolute error) as it was raised. We decided to develop three versions of the ensemble: each with 1000 trees, but varying the the number of features (K) from 100 to 300 to 500. Each ensemble was then trained and evaluated on the exact same data as the LSTM model, allowing for direct comparisons. The ensembles produce confidence intervals based on the percentile method ($\alpha = 0.05$).

2.7 Evaluation

To compare the various models we developed, a once-off validation was performed. Identical data augmentation and preprocessing was performed for each model and they each saw the identical sets of sub-samples in training and validation – allowed the results to be directly compared.

Dependent and Independent Variables The *independent* variable in our experiment was the model type (and select hyperparameters) used. The values for this variable were two variations on the LSTM model—one with fixed CI sampling and one with dynamic CI sampling—and three variations on the random forest—with varying numbers of features (K) being selected. We opted for only a single independent variable (with a limited number of values) to constrain the scope of the experiment to investigate our hypotheses about confidence intervals (CIs). We had already used techniques like grid search during model development to optimise hyperparameters, so we decided to keep those variables constant once robust values had been found. There were three *dependent* variables:

1. **MAE**: the mean absolute error for each model,
2. **Mean CI width**: the mean difference in the confidence intervals for each model,
3. **Accuracy**: the total accuracy of each model. Accuracy was measured as the fraction of validation samples where the true HR value fell within the confidence interval (exclusively).

We chose to track these three variables because they allowed us to make holistic conclusions about our different models, their strengths, and their weaknesses. Given the clinical context of this project, accurate predictions alone were not sufficient. MAE allowed us to quantify how good the models were on average at predicting the true value. The accuracy metric allowed us to quantify how effective the models were at determining their confidence. But CIs are only useful if they are not unhelpfully wide. The metric of mean CI width allowed us to compare the variations in model confidence.

Protocol Evaluation was performed with an 80:20 train-test split on the 779 augmented sub-samples. A fixed random seed of 55 was used when shuffling the data prior to splitting. This ensured that each model received identical samples in training and validation, and allows other researchers to replicate our results.

The K value (number of features to select) was varied for the random forest models, whilst all other hyperparameters were fixed. For the LSTM models, the size of the distributions was allowed to vary in the dynamic model, but was a constant of 1000 in the fixed model. By setting the size of the LSTM sample distribution directly proportional to the data quality metrics, data quality was able to influence the dynamic model's estimates. This encouraged the model to balance between the precision of the intervals and uncertainty estimation. Both the LSTM and random forest models produced a distribution of values for each validation case and the confidence intervals were calculated using the percentile method ($\alpha = 0.05$). The aggregated predictions were determined by the means of the distributions for both kinds of architecture, as is standard with bagging [1].

3 Results and Discussion

The results of evaluation across all models are presented in Table 3.

The overall best-performing model was the LSTM with dynamic sampling based on signal quality (**LSTM-dynamic**). Our implementation was able to utilise the bespoke signal quality functions we developed to regulate its confidence intervals. It was able to achieve the highest accuracy (68.5%), whilst maintaining the second most narrow confidence intervals. Whilst it had the highest mean absolute error of all the models, the differences were marginal. The LSTM with fixed sample sizes had very low accuracy (21.1%), but this makes sense when observing that its confidence intervals were narrower on average than its error margin.

The dramatic difference in performance between the fixed- and dynamic-sampling with LSTMs is notable, as it highlights the need for confidence estimation to be a deliberate design choice – especially in high-risk medical tasks. If we had compared the LSTM implementations only in terms of mean error, they would have been virtually indistinguishable. However, it is clear from the other metrics that **LSTM-dynamic** is vastly more useful as a decision-support sys-

tem. The final model architecture is presented in Fig. 6 and the hyperparameters are detailed in section 2.5.

The random forest (RF) models saw increased accuracy as the number of selected features (K) was increased, with **RF-500** containing the true value within its CIs in 67.3% of cases. However, the added dimensionality also increased the average error and the width of the CIs considerably. It appears that the addition of features above $K = 100$ simply resulted in more extreme predictions at the tails of the distribution. Despite this, the RF models handily outperformed the LSTM models in terms of absolute error.

Table 3. Comparison of heart rate estimation performance across 5 models, with the best overall performer emboldened. Accuracy is the fraction of labels that fell within the confidence intervals. CI = confidence interval (95%). MAE = mean absolute error. RF-K = random forest ensemble with K-best features. The structure and hyperparameters of the best model are shown in Fig. 6.

Model	**Accuracy (%)**	**Mean CI width**	**MAE**
LSTM-fixed	12.1	2.72	3.09
LSTM-dynamic	**68.5**	**8.46**	**3.13**
RF-100	32.7	17.89	2.36
RF-300	56.4	20.28	2.50
RF-500	67.3	24.31	2.70

These results suggest that a dynamically-sampling LSTM architecture is best for quantifying uncertainty (through confidence interval estimates) on this dataset. As more data (and a more uniform distribution of values) comes in through the *CoVital* project, the performance is likely to increase further, as is typical of deep neural networks [7]. However, given the current limits in the size of data, a random forest ensemble had the lowest prediction error. It is possible that a combination of these two approaches could be used to yield both low prediction error *and* precise confidence intervals, which is an interesting area for future research.

4 Other Approaches and Lessons Learned

Multiple algorithms were considered as partial alternatives to our Deep Learning approach. The most promising method for time series regression was utilising Echo-State networks (ESN) and exploiting an efficient Bayesian optimisation process to quantify continuous blood pressure. Compared to other recurrent architectures, the ESN offers a faster training time and lower computational constraints, which may allow the deployment on an embedded monitoring device [5]. Another interesting approach for quantifying uncertainty would be in applying Kernel Mixture or Density Mixture networks for continuous probability distribution estimation. The former uses kernels as a base for the estimation and the latter evaluates the mixture of multiple distributions.

With our approach, however, we opted for the most flexible model in earlier stages of the project to obtain preliminary results and build from there. This choice proved useful as we had to revise our project structure due to limitations in the available data. With our hybrid LSTM, we were able to do this with minimal changes.

Through this project, we learned how to integrate research findings from multiple domains and extend them to produce effective code for preprocessing, evaluating signal quality, and generating confidence estimates. We also extended our practical knowledge of deep neural networks and ensemble techniques, whilst picking up new skills in signal processing for time series data. In particular, the knowledge obtained from applying time-frequency domain transformations and using the associated filters will likely be valuable to us in future projects involving periodic data.

Most importantly, we gained valuable experience in overcoming the challenges of limited real-world datasets. By maximising our options through researching other datasets and experimenting with different target features, we were able to salvage the underlying value in our methodology despite the limitations of the current *CoVital* dataset. Given that there are only 67 samples in the raw dataset, we were truly surprised to obtain such robust results from a deep neural network, as they typically require enormous datasets.

5 Conclusions

The aim of this project was to evaluate different techniques for uncertainty estimation on the real-world *CoVital* dataset. Initially, we hoped to use blood oxygen saturation (SpO2) as our target feature because of its relevance to Covid-19 physiology. Unfortunately, there was an insufficient amount of clinical data available at the time, and we instead used the more uniformly-distributed heart rate measurements to validate our techniques. Given that the underlying data and preprocessing pipeline were identical, we believe that our results will generalise to SpO2 estimation when sufficient data is available.

We extended previous work in the literature to develop a bespoke preprocessing pipeline and signal quality indexes, both of which were instrumental in developing robust models with useful confidence intervals. We investigated the use of dropout in an LSTM architecture and compared performance with a random forest ensemble in terms of three metrics. The best overall performer was an LSTM architecture that varied the size of the dropout samples depending on the signal quality scores we developed.

Uncertainty estimation is of vital importance to learning tasks in the medical domain. As more data becomes available, we hope that our findings can assist the *CoVital* project in developing safe, robust models that make a positive impact during the global pandemic.

References

1. Breiman, L.: Random Forests. Tech. rep. (1999)
2. Cheng, S.C., Chang, Y.C., Fan Chiang, Y.L., Chien, Y.C., Cheng, M., Yang, C.H., Huang, C.H., Hsu, Y.N.: First case of Coronavirus Disease 2019 (COVID-19) pneumonia in Taiwan. Journal of the Formosan Medical Association **119**(3), 747–751 (mar 2020). https://doi.org/10.1016/j.jfma.2020.02.007
3. Ching, T., Himmelstein, D.S., Beaulieu-jones, B.K., Kalinin, A.A., Do, B.T., Way, G.P., Ferrero, E., Agapow, P.m., Zietz, M., Hoffman, M.M., Xie, W., Rosen, G.L., Lengerich, B.J., Israeli, J., Lanchantin, J., Woloszynek, S., Carpenter, A.E., Shrikumar, A., Xu, J., Cofer, E.M., Lavender, C.A., Turaga, S.C., Alexandari, A.M., Lu, Z., Harris, D.J., Decaprio, D., Qi, Y., Kundaje, A., Peng, Y., Wiley, L.K., Huang, A., Gitter, A., Greene, C.S.: Opportunities and obstacles for deep learning in biology and medicine. The Royal Society (2018). https://doi.org/10.1098/rsif.2017.0387
4. Elgendi, M.: Optimal signal quality index for photoplethysmogram signals. Bioengineering **3**(4), 1–15 (2016). https://doi.org/10.3390/bioengineering3040021
5. Franco, G., Cerina, L., Gallicchio, C., Micheli, A., Santambrogio, M.D.: Continuous blood pressure estimation through optimized echo state networks. In: Tetko, I.V., Kůrková, V., Karpov, P., Theis, F. (eds.) Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions. pp. 48–61. Springer International Publishing, Cham (2019)
6. Helpful Engineering: CoVital Project, `https://www.covital.org/`
7. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation **18**(7), 1527–1554 (may 2006). https://doi.org/10.1162/neco.2006.18.7.1527
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735, `https://doi.org/10.1162/neco.1997.9.8.1735`
9. Karlen, W., Lim, J., Ansermino, J.M., Dumont, G., Scheffer, C.: Design challenges for camera oximetry on a mobile phone. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS pp. 2448–2451 (2012). https://doi.org/10.1109/EMBC.2012.6346459
10. Lamonaca, F., Carni, D.L., Grimaldi, D., Nastro, A., Riccio, M., Spagnolo, V.: Blood oxygen saturation measurement by smartphone camera. 2015 IEEE International Symposium on Medical Measurements and Applications, MeMeA 2015 - Proceedings pp. 359–364 (2015). https://doi.org/10.1109/MeMeA.2015.7145228
11. MIT Technology Review: Google's medical AI was super accurate in a lab. Real life was a different story. (2020), `https://www.technologyreview.com/2020/04/27/1000658/google-medical-ai-accurate-lab-real-life-clinic-covid-diabetes-retina-disease/`
12. Nemcova, A., Jordanova, I., Varecka, M., Smisek, R., Marsanova, L., Smital, L., Vitek, M.: Monitoring of heart rate, blood oxygen saturation, and blood pressure using a smartphone. Biomedical Signal Processing and Control **59**, 101928 (2020). https://doi.org/10.1016/j.bspc.2020.101928, `https://doi.org/10.1016/j.bspc.2020.101928`
13. Nemcova, A., Jordanova, I., Varecka, M., Smisek, R., Marsanova, L., Smital, L., Vitek, M.: Monitoring of heart rate, blood oxygen saturation, and blood pressure using a smartphone. Biomedical Signal Processing and Control **59**, 101928 (may 2020). https://doi.org/10.1016/j.bspc.2020.101928

14. Tayfur, I., Afacan, M.A.: Reliability of smartphone measurements of vital parameters: A prospective study using a reference method. The American Journal of Emergency Medicine **37**(8), 1527–1530 (2019). https://doi.org/https://doi.org/10.1016/j.ajem.2019.03.021, http://www.sciencedirect.com/science/article/pii/S0735675719301706
15. Wang, E.J., Li, W., Zhu, J., Rana, R., Patel, S.N.: Noninvasive hemoglobin measurement using unmodified smartphone camera and white flash. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS pp. 2333–2336 (2017). https://doi.org/10.1109/EMBC.2017.8037323
16. Welch, P.: The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. IEEE Transactions on audio and electroacoustics **15**(2), 70–73 (1967)