Continuous Blood Pressure estimation using traditional ML and DL techniques

Group members: Aruzhan Suleimenova, Baurzhan Kumarioldanov.
Adviser: Jurn Gyu Park

Executive Summary (10%)

This project aims to enable **continuous, non-invasive blood pressure (BP) estimation** using physiological signals—PPG, ECG, and PCG—through a combination of **deep learning and traditional machine learning models**. Motivated by the need for real-time cardiovascular monitoring, the work addresses a critical healthcare challenge with potential applications in **smartwatches and edge devices**. The proposed solution follows a dual-track approach: one using **raw signal data**, and another using **feature-engineered data**. For the raw data track, various preprocessing techniques—including butterworth filtering, moving average smoothing, and discrete wavelet transform—were evaluated, with the best results achieved by an **advanced LSTM model** trained on PPG+ECG signals filtered with butterworth+MAF (4.03 mmHg for SBP, 2.27 mmHg for DBP). In the feature-engineered track, statistical and physiological features were extracted, and models like RF, GBR, LSTM, and a Transformer inspired by the **TransfoRhythm** architecture were tested, with the RF Regressor delivering the best performance (MAE of 1.2 mmHg for SBP and 0.3 mmHg for DBP). The project also explored **ABP prediction from raw PCG signals** on basic CNN, Transformer and LSTM based models, extending the scope to multimodal signal fusion. CNN achieved the lowest MAE of 13.19 mmHg .

This project aligns closely with the principles of a **computing-based solution** by addressing a significant healthcare problem through a structured technical approach. The **design phase** involved defining a dual-track architecture to handle raw and feature-engineered physiological data, incorporating signal processing techniques and model configurations. In the **implementation phase**, we developed end-to-end pipelines that preprocess signals, extract meaningful features, and train predictive models using established frameworks. Finally, the **evaluation phase** included benchmarking multiple models (e.g., LSTM, Transformer, RF) across different signal combinations using metrics such as MAE and RMSE, allowing for a comprehensive assessment of model accuracy and system effectiveness in estimating continuous blood pressure.

Introduction (10%)

Problem, Motivation, and Significance

Hypertension and cardiovascular diseases are among the leading causes of mortality worldwide. Despite their prevalence, continuous and non-invasive blood pressure (BP) monitoring remains a significant challenge. Traditional methods rely on cuff-based devices, which are not suitable for real-time, continuous use, especially outside of clinical settings. This creates a critical gap in early detection and preventive care.

The motivation behind this project stems from the need to develop a **non-invasive, continuous, and wearable BP monitoring system** that can operate in real-time using **widely available physiological signals** such as PPG, ECG, and PCG. By enabling seamless integration into devices like smartwatches and edge sensors, the proposed solution has the potential to **transform preventive healthcare** and significantly reduce the risk of cardiovascular events.

Proposed Solution

To address this problem, the project proposes a **dual-pipeline framework** for BP estimation:

1. A **raw signal-based pipeline**, which processes unfiltered physiological signals through advanced deep learning architectures, including U-Net, LSTM, and Transformer models.

2. A **feature-engineered pipeline**, which leverages both statistical and physiological signal features as input to machine learning and deep learning models like Random Forest, Gradient Boosting Regressor, and Transformer-based models.

Each pipeline undergoes **comprehensive preprocessing**, feature extraction (if applicable), and model evaluation across different signal combinations (e.g., PPG only, ECG only, PPG+ECG). The solution also explores **ABP prediction from heart sound (PCG) data**, making it a truly **multimodal approach** to physiological signal processing.

Report Organization

The remainder of this report is organised as follows:

- **Section 2**: Describes the background and related work.
- **Section 3**: Provides a detailed description of the solution, dataset sources, preprocessing techniques, and feature extraction methods.
- **Section 3**: Details the execution process, architecture and training configurations of the deep learning and machine learning models.

- **Section 4**: Presents experimental results, comparisons across models, and evaluation metrics.
- **Section 5**: Discusses challenges, limitations, and opportunities for future work.
- **Section 6**: Concludes the report by summarizing the key findings and implications for real-world deployment.

2. Background and Related Work (15%)

Continuous, cuffless blood pressure (BP) estimation has been tackled along three major lines: **(1) ECG‑only models, (2) PPG‑only models, and (3) multimodal ECG + PPG models,** with the addition of lesser explored **PCG-only models**.
Multimodal classical ML studies such as Kachuee et al. [1, 2] and a wearable‑device study [24] predict BP from pulse‑transit‑time (PTT) features extracted from ECG–PPG pairs, typically with tree ensembles or linear regressors. More recent *deep‑learning* works (Lo et al. [3], Yen et al. [25], Kamanditya et al. [26]) remove hand-crafted features and use LSTM or hybrid CNN–LSTM networks on windowed segments from the MIMIC‑II/III databases, reporting lower mean errors but at the cost of larger models and heavier preprocessing.

For **PPG‑only** estimation, U-Net variants dominate: PPG2ABP [4] and Athaya et al. [27] generate coarse ABP waveforms and then refine them, while BP‑Net [5] adds self‑supervised pre‑training to improve generalisation. LSTM‑based auto‑encoders with transfer learning (Harfiya et al. [29]) and RNN/attention hybrids (El‑Hajj et al. [30]) incorporate derivative‑based features, whereas Transformer architectures push the state‑of‑the‑art in sequence modeling (ArterialNet [32], STP [6]). The first PPG-only study on the **new MIMIC‑IV** waveform set—TransfoRhythm [13]—combines cycle‑level feature engineering with Transformers.

ECG‑only solutions are less common: a stacking framework on handcrafted‑complexity features (Simjanoska et al. [7]), a random‑forest regressor with RFE‑selected features (Wuerich et al. [8]), and a microcontroller-based ANN demonstrator (Syah et al. [9]). These show feasibility but generally lag in multimodal or PPG‑only accuracy.

Solutions relying solely on PCG signals remain extremely rare. While PCG provides valuable cardiovascular information, it is typically paired with ECG or PPG signals to extract surrogate features for accurate BP estimation, which was the case with the study by Dastjerdi et al. [16] that utilized PCG in conjunction with ECG to estimate PTT. Pure PCG-based approaches are still in their infancy and face significant performance limitations. A recent attempt by Kokkhunthod et al. [17] proposed a CNN-based regression model using only PCG signals, but the results were notably higher than most ECG+PPG or even PPG-only methods, indicating that PCG-only models currently fall short in estimation accuracy.

Gap & rationale for our methodology.
(1) No prior work evaluates both *raw* and *feature-engineered* pipelines side-by-side on MIMIC-IV;
(2) multimodal ECG + PPG deep models on this latest dataset remain unexplored;
(3) existing studies often lock into a single preprocessing scheme, making it hard to separate filtering gains from model gains.
To fill these gaps, we **designed a dual-track framework**: raw-signal models (U-Net, advanced LSTM, Transformer) tested under four filtering strategies, and feature-engineered models (RF, GBR, LSTM, TransfoRhythm-style Transformer) that systematically combine statistical and physiological features. By benchmarking the two tracks across identical train/validation splits, we can isolate the contribution of preprocessing, modality choice, and model family—thus providing a comprehensive computing-based analysis absent from current literature.

3. Project Approach (20%)

Our solution is organized as a **dual-pipeline architecture**—a *Raw-Signal Pipeline* and a *Feature-Engineered Pipeline*—that share a common data-ingestion layer and evaluation backend (Fig. 1).

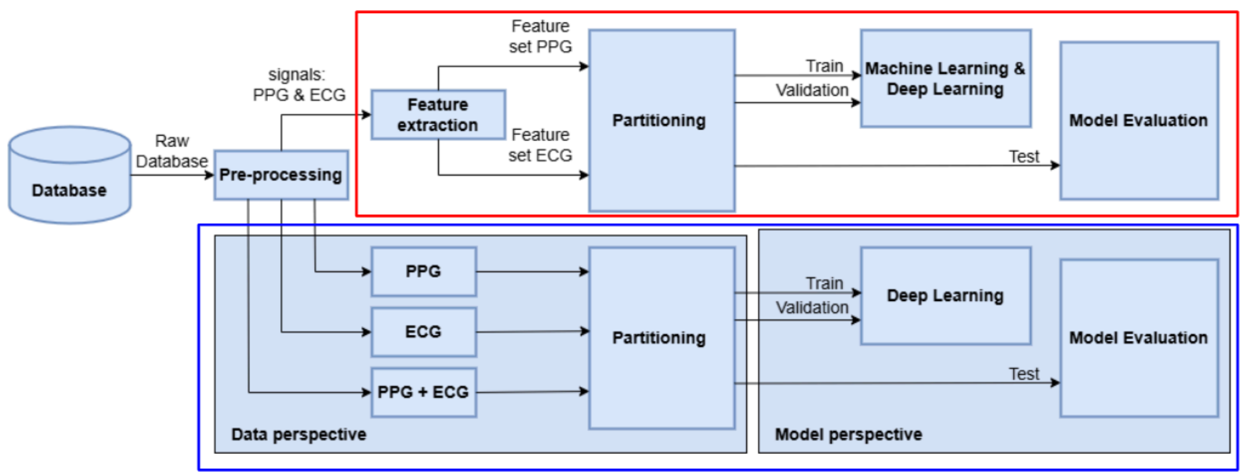

Figure 1. Overview of End-to-End Pipelined Framework from Database to Model Evaluation. [1]

3.1 Software / Hardware Architecture

Data Layer Raw ECG, PPG and ABP were streamed from the **MIMIC-IV waveform database** with **WFDB-Python**. We retained recordings > 10 min and discarded traces containing NaNs or implausible ABP values (outside the Kachuee et al. physiological range).

Pre-processing Layer

The four filtering paths run on a **12‑core Intel Core i7 workstation**: 1) Butterworth (5th order, 0.5–60 Hz), 2) Butterworth + moving‑average (MAF), 3) Discrete Wavelet Transform (Daubechies 9) via PyWavelets, 4) NeuroKit's *clean‑PPG* and *clean‑ECG* routines.

For the raw‑signal track, we compared all four; Butterworth+MAF and DWT were selected as the two "leading" filters.

For the feature-engineered track, we used 2 filtering techniques: Butterworth and Butterworth+MAF.

Model Layer

Compute Resources. We split training workloads to match their hardware needs: all deep‑learning experiments (U‑Net, advanced LSTM, Transformer) were trained on a **Tesla P100 GPU with 16 GB VRAM** provided by Kaggle kernels, while the tree‑based models (Random Forest and Gradient Boosting Regressor) and feature‑engineering tasks ran on our local **12‑core Intel Core i7 workstation**.

- **Raw‑Signal models**:
    - Baseline U‑Net and LSTM, advanced BP-Net and LSTM and a vanilla Transformer-encoder. Each was trained on ECG only, PPG only, and PPG+ECG inputs.
    - Basic Transformer-encoder, LSTM and CNN models were trained on PCG input.
- **Feature‑Engineered models**: RF, GBR, LSTM, and a Transformer that follows the **TransfoRhythm** blueprint. Features include statistical moments, morphology (dicrotic‑notch timing, PTT), RR‑interval variability, and SDPPG amplitudes. Six feature + filter combinations were benchmarked.

3.2 Algorithms & Workflows

1. **Feature extraction: 1) PPG signals -** We adopt the *TransfoRhythm* workflow [4]**,** extracting features from both the raw PPG waveform and its second derivative (SDPPG). 2) **ECG signals -** Following the methods in [6], [7], [8], we compute two statistical feature sets (mean, quartiles, standard deviation, etc.), and physiological features such as R‑peak locations and QRS‑complex intervals. These are implemented with NumPy, the Pan–Tompkins algorithm, and functions from `scipy.signal`. This dual‑track extraction yields complementary statistical and physiological descriptors for subsequent modelling.
2. **Pipeline Selection:** Local optimum search over four filters × two base DL models, ranked by validation errors → pick top‑2 filters.
3. **Hyperparameter Tuning.** For both pipelines, we adopted a *local‑optimum* strategy: each tuning run fixed all but one or two key hyperparameters (e.g., learning rate and hidden‑size for LSTM, tree‑depth for RF) and searched their optimal values while holding the remaining settings constant. This controlled, one-factor-at-a-time approach isolated

each parameter’s impact and allowed us to converge on the best configuration without exploding the search space.

3.3 Third-Party Components & Integration

Component	Why did we use it	Integration
WFDB-Python	Access to MIT-format waveforms	Incorporated as a part of the data cleaning stage, where we extracted time-series data and applied initial filtering.
PyWavelets	for DWT filtering	The library was integrated into the preprocessing pipeline for DWT filtering.
Scipy.signal	for Butterworth filtering	The library was integrated into the preprocessing pipeline for BW and BW+MAF filtering.
NeuroKit	Proven bio-signal filters	The library was integrated into the preprocessing pipeline for neurokit filtering.
Weights & Biases	Reproducible experiment logging	Auto-callbacks record metrics, configs, and GPU stats

---

3.4 Team Roles & Collaboration

Name	Role	Key Contributions
Aruzhan	**Raw data pipeline Lead**	Preprocessing of raw data, selection and evaluation of raw data-based models.
Baurzhan	**Feature-engineered pipeline Lead**	Feature extraction from preprocessed data, selection and evaluation of feature-engineered models.

We met **weekly on Thursdays** to plan and review progress with our supervisor, while day-to-day coordination took place in **Telegram**, and task tracking and document sharing were managed in **Notion**. This lightweight cadence kept communication clear and progress visible, allowing the team to iterate quickly yet stay aligned with project goals.

Project Execution (15%)

Project Progress and Key Decisions (Last Two Semesters)

Over the last two semesters, our project evolved significantly across both the raw and feature-engineering pipelines. We began with raw signal processing, where the initial challenge was selecting appropriate filtering techniques for ECG and PPG signals. We experimented with multiple filtering methods, facing difficulties with signal ranges and noise. To address this, we decided to adopt filtering practices commonly used in similar datasets such as MIMIC-II and MIMIC-III, which provided more stable and standardized preprocessing.

Model selection was another major challenge. Many research papers on blood pressure prediction lack publicly available or reproducible code, which made it difficult to replicate their reported results. In some cases, we found reproducible models; however, even those yielded different results, likely due to differences in library versions, training environments, or other implementation details. For models without code, we reconstructed them based on descriptions provided in the papers. This process, while insightful, introduced potential deviations from the original setups, which may have affected performance.

Training the models on raw signals also presented hardware-related limitations. Initially, we used a lab machine with a 5GB GPU. As we moved on to LSTM and Transformer-based models that required overlapping windows for sequence generation, the memory demand exceeded our system's capacity. To address this, we transitioned to Kaggle's 16GB GPU environment. However, this introduced its own challenges: session time limits (12 hours) and occasional freezing, both of which disrupted training. For example, when training an LSTM on a combined PPG+ECG input, we encountered out-of-memory (OOM) issues due to the sequential nature of LSTMs. To mitigate this, we split the dataset into two parts and trained the model sequentially.

In contrast, the feature-engineering pipeline primarily relied on classical ML models that ran efficiently on CPUs, so training itself did not pose significant problems. The main challenge here was in the creation of the feature-engineered dataset. As with the modeling phase, most papers did not provide code for feature extraction. We had to manually analyze papers and implement the feature extraction techniques ourselves. Moreover, the ECG and PPG signals from the ICU dataset had variable periodicity across subjects, complicating cycle-based feature extraction. To address this, we designed parameterizable feature extraction logic that adapts to the signal's structure, effectively reducing error rates in cycle detection and improving robustness.

Teamwork and Collaboration

Our team adopted a flexible and collaborative approach to task distribution. Initially, Aruzhan focused on building the feature-engineering pipeline and developed a semi-automated toolbox for dataset creation. Baurzhan began with the raw signal pipeline, where he tested different preprocessing methods and experimented with various model architectures.

After initial development, we decided to swap roles to gain a deeper understanding of each other's work and bring fresh perspectives to both pipelines. Aruzhan took over the raw pipeline, testing more advanced deep learning architectures and refining the preprocessing strategy. Meanwhile, Baurzhan expanded the feature-engineering pipeline by integrating more complex and domain-specific features.

Throughout the project, we held regular meetings to discuss progress, identify roadblocks, and brainstorm solutions together. Decision-making was collaborative, and we often worked together to debug difficult issues or review each other's code. This shared responsibility helped ensure continuity in the project and promoted mutual learning. Leadership was shared based on task ownership and technical expertise, allowing each member to take initiative at different phases.

Evaluation (20%)

To evaluate whether our project effectively addressed the problem of continuous blood pressure prediction, we used several well-established metrics commonly applied in this research area. These included:

- **Mean Absolute Error (MAE)**
- **Root Mean Squared Error (RMSE)**
- **Mean Error ± Standard Deviation (ME ± SD)**
- **British Hypertension Society (BHS) Standard**

The use of MAE and RMSE helped us quantify how close the model predictions were to actual blood pressure values, while ME ± SD gave us insights into prediction consistency and systematic bias. Importantly, we used the BHS standard to assess whether our models met clinical accuracy requirements, which allowed us to evaluate the real-world applicability of our solution in healthcare settings.

In addition to accuracy, we assessed **training time and computational efficiency**, since deployment in real-time or resource-constrained environments (e.g., wearable devices) requires both accuracy and speed. These trade-offs were important in our comparison of traditional machine learning and deep learning approaches.

PreP.	Error		U-Net			BP-Net			LSTM-stan.			LSTM-adv.			Transformer		
			ECG only	PPG only	PPG+ECG	ECG only	PPG only	PPG+ECG	ECG only	PPG only	PPG+ECG	ECG only	PPG only	PPG+ECG	ECG only	PPG only	PPG+ECG
BW+MAF (PP 2)	MAE	SBP	8.09	9.16	6.42	7.56	7.36	6.50	5.15	8.23	5.15	4.46	6.72	**4.03**	6.85	9.35	6.80
		DBP	3.12	3.52	2.89	4.62	3.85	3.53	1.52	2.61	**1.21**	2.44	3.62	2.27	3.68	4.53	3.53
		MAP	5.02	6.03	4.16	5.49	5.99	5.14	3.24	6.11	3.04	3.27	6.16	**3.02**	4.62	7.93	4.30
	RMSE	SBP	11.42	13.05	9.06	10.10	10.34	9.04	7.05	10.74	7.18	6.66	9.86	**6.26**	9.42	12.71	9.14
		DBP	4.84	5.39	4.80	6.08	5.35	5.05	1.74	2.91	**1.44**	4.29	5.52	4.16	5.60	6.44	5.35
		MAP	7.49	9.24	6.35	7.95	8.79	7.64	5.15	9.90	5.06	5.20	9.99	**5.02**	6.85	11.58	6.36
	ME±SD	SBP	4.41 ± 10.54	6.39 ± 11.38	4.02 ± 8.12	-0.82 ± 10.06	1.59 ± 10.22	0.06 ± 9.04	-0.56 ± 7.03	-2.61 ± 10.42	0.23 ± 7.18	1.22 ± 6.55	-1.50 ± 9.74	**0.94 ± 6.19**	3.52 ± 8.73	1.73 ± 12.59	4.61±7.89
		DBP	-0.47 ± 4.82	-0.27 ± 5.38	-0.62 ± 4.76	-3.64 ± 4.87	-2.32 ± 4.82	-2.06 ± 4.61	1.38 ± 1.06	2.52 ± 1.46	**0.99 ± 1.05**	-0.62 ± 4.24	0.84 ± 5.46	-0.62 ± 4.12	-2.30 ± 5.10	0.50 ± 6.42	-2.34±4.81
		MAP	0.56 ± 7.47	0.22 ± 9.24	0.0036 ± 6.3540	0.68 ± 7.92	-0.10 ± 8.79	-0.78 ± 7.60	3.24 ± 4.00	6.11 ± 7.79	3.04 ± 4.04	3.27 ± 4.04	6.16 ± 7.87	**3.02 ± 4.00**	-0.45 ± 6.84	0.39 ± 11.57	0.05±6.36
	BHS	SBP	C	Fail	B	C	C	B	B	Fail	A	A	B	A	C	Fail	C
		DBP	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
		MAP	A	B	A	B	B	B	A	B	A	A	B	A	A	C	A
Training time			4.5h	4.5h	4.5h	3h	3h	3h	6.9h	6.9h	7.8h	6.9h	6.9	7.8h	7.7h	7.6h	6.1h
DWT (PP 3)	MAE	SBP	6.65	7.35	5.55	5.81	6.23	5.59	6.32	7.59	6.09	5.68	6.27	**4.64**	6.94	8.46	6.65
		DBP	3.07	3.58	3.00	3.15	3.03	3.36	**1.09**	2.78	1.27	3.01	3.49	2.60	3.50	4.00	3.21
		MAP	4.51	5.77	4.00	4.46	5.30	4.46	3.95	6.02	3.78	4.29	5.98	**3.56**	4.57	6.45	4.34
	RMSE	SBP	9.50	10.89	8.16	8.52	9.01	7.94	8.19	10.33	8.10	8.01	9.35	**6.75**	9.51	11.60	9.01
		DBP	4.79	5.69	4.95	4.74	4.58	4.76	**1.35**	2.99	1.47	4.86	5.30	4.43	5.31	5.90	4.99
		MAP	6.76	8.98	6.20	6.76	8.01	6.68	6.00	9.64	5.90	6.52	9.56	**5.59**	6.76	9.58	6.37
	ME±SD	SBP	2.58 ± 9.15	3.80 ± 10.21	2.82 ± 7.66	0.87 ± 8.48	-0.12 ± 9.01	-0.08 ± 7.94	-3.21 ± 7.54	-2.71 ± 9.97	-1.91 ± 7.87	-1.10 ± 7.93	-0.94 ± 9.30	**-0.24 ± 6.74**	4.03 ± 8.62	3.36 ± 11.10	4.08 ± 8.03
		DBP	-0.30 ± 4.78	-1.44 ± 5.50	-1.57 ± 4.70	-1.79 ± 4.39	-0.94 ± 4.48	-2.10 ± 4.27	**0.50 ± 1.25**	2.73 ± 1.23	1.09 ± 0.99	-0.66 ± 4.81	0.92 ± 5.22	-0.61 ± 4.39	-1.71 ± 5.03	-0.35 ± 5.89	-1.30 ± 4.82
		MAP	-0.0015 ± 6.7594	-1.31 ± 8.88	-0.65 ± 6.17	-0.45 ± 6.74	-0.40 ± 8.00	-0.93 ± 6.61	3.95 ± 4.53	6.02 ± 7.53	3.78 ± 4.53	4.29 ± 4.90	5.98 ± 7.45	**3.56 ± 4.31**	0.02 ± 6.76	0.71 ± 9.56	0.32 ± 6.36
	BHS grade	SBP	B	C	B	B	B	B	C	C	B	B	B	A	C	C	C
		DBP	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
		MAP	A	B	A	A	B	A	A	B	A	A	B	A	A	B	A
Training time			4.5h	4.5h	4.5h	3h	3h	3h	6.9h	6.9h	7.8h	6.9h	6.9	7.8h	7.7h	7.6h	6.1h

Table 3. Test Results on all types of raw data. (Best and Worst in each Error. Highlighted for the Best on both BW+MAF and DWT Datasets.)

	RF	GBR	LSTM	Transformer (seq2seq)	TransfoRhythm Architecture
MAE SBP	1.2	1.334	1.90	1.3897	1.7642
MAE DBP	0.484	0.323	0.45	0.7392	0.4998
RMSE SBP	1.714	2.014	2.43	1.8315	2.3727
RMSE DBP	0.691	0.427	0.66	0.9621	0.6777
BHS Grade for SBP	A	A	A	A	A
BHS Grade for DBP	A	A	A	A	A

Table 4. Test results for FE data on BW+MAF preprocessing. Best results are highlighted.

Our primary focus was on ECG-based models, given that ECG is often underexplored in the context of blood pressure prediction, despite its increasing availability in consumer-grade devices like smartwatches. To ensure a fair and comprehensive evaluation, we also tested and compared model performance using:

- **PPG-only signals**
- **ECG-only signals**
- **Combined PPG + ECG signals**

This allowed us to evaluate how much additional value multimodal data integration could offer. The results showed that models performed best when trained on **PPG + ECG**, which aligns with our hypothesis that combining complementary signals improves prediction accuracy due to richer physiological information. The results can be seen in Table 3 for models that used raw data.

Model	CNN	Transformer	LSTM
MAE (mmHg)	13.1866	14.0131	13.5713
RMSE (mmHg)	17.3089	17.5759	16.9540

Table 5. Test results for models that used PCG-only dataset. Best results are highlighted.

The results of the feature-engineered data can be found in Table 4. This time, due to the abovementioned under-exploration of ECG-only methods, we considered only ECG signals and 30 features extracted from it.

The PCG data results can be seen in Table 5. The errors are considerably higher than the ones yielded by PPG, or ECG data. The possible reasons for that include small dataset, unavailability of prior research in this field, and low correlation between ABP and PCG.

Data Collected & Brief Analysis

Our evaluation used a preprocessed dataset from MIMIC-IV, consisting of synchronized PPG and ECG signals. We segmented this data into overlapping windows and extracted both raw and feature-based representations for each cycle. We evaluated model performance using cross-validation and tested models on unseen data to ensure generalization. PCG data was collected from an open-source database made by Rafael Gonzalez - Landaeta, Brenda Ramirez & Jose Mejia [18] containing PCG signals and SBP.

Key observations for Raw Data:

- **PPG-only models** were less consistent and struggled with certain waveform variability.
- **ECG-only models** had decent performance and showed promise, especially with more advanced architectures.
- **PPG + ECG models** consistently achieved lower MAE/RMSE and met or came close to BHS standards, suggesting strong clinical potential.

Key observations for FE Data:

- **Feature Engineering is highly effective** as all models achieved BHS Grade A.
- **Random Forest Regressor** showed the best results for SBP prediction.
- **Gradient Boosting Regressor** showed the best results for DBP prediction, meaning ML models outperform DL models when extracted features are given.

If we look at PCG data, we can immediately see that the results are unacceptable according to the BHS standard. There is a big room for improvement in terms of both dataset size and model optimization.

Though we didn’t conduct a user-based evaluation (e.g., with clinicians), our use of established clinical standards like BHS allowed us to simulate real-world performance expectations.

Conclusion and possible future work (5%)

Key Findings and Contributions

Throughout the project, we explored various machine learning and deep learning approaches for continuous blood pressure prediction using PPG and ECG signals. Our findings can be summarized as follows:

- **LSTM-based models** achieved the **best overall performance**, with the lowest **Mean Absolute Error (MAE) of 4.03 for Systolic Blood Pressure (SBP)** and **1.09 for Diastolic Blood Pressure (DBP)**. These results demonstrate the LSTM’s effectiveness in capturing temporal dependencies in physiological signals.
- However, the **training time for LSTM models was relatively high**, ranging from **7 to 8 hours**, highlighting a need for optimization in model complexity and training efficiency.
- **CNN-based models** also showed **competitive performance**, especially when trained on **features extracted via Discrete Wavelet Transform (DWT)**. Notably, these models required significantly **less training time (~3 hours)** while still producing promising results, suggesting their suitability for resource-constrained applications.
- **Transformer-based models** delivered **stable but moderate results**. While they show potential due to their capacity for parallel processing and long-range feature modeling, **further refinement is needed** to fully exploit their strengths in capturing complex signal features.
- Simple **ML models** also showed competitive performance and even outperformed DL models if **extracted features** are given.

- Having already **extracted features lessens the importance of temporal dependencies** making other models compatible with LSTM that outperformed other models on raw data.
- **PCG-only models** performance is **unsatisfactory**, but they require further research before making conclusions on usability of such models, due to the small size of the used dataset and absence of model optimization.

Future Enhancements and Areas for Improvement

Based on our findings, we identify several potential directions for future work:

- **Optimize LSTM models** by exploring lightweight architectures (e.g., GRU, BiLSTM with attention) or using model compression techniques to reduce training time without compromising performance.

- **Refine CNN architectures**, particularly by experimenting with hybrid models that combine CNN feature extraction with lightweight sequence modeling, aiming to balance accuracy and efficiency.

- **Improve Transformer models** by adjusting architecture parameters (e.g., attention heads, depth, input representation) and incorporating domain-specific positional encoding schemes to better align with physiological signal patterns.

- **Extend evaluation** to include real-time constraints and potential integration with wearable devices, ensuring practicality for real-world deployment.

- **Incorporate user feedback**, such as from clinicians or end-users, to guide model improvement and usability assessment in clinical or home-monitoring settings.

- **Collect a larger PCG dataset** that contains both continuous PCG and ABP signals for better generalization of training models.

References (5%)

Implementation could be found here: link

[1] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, “Cuffless blood pressure estimation algorithms for continuous health-care monitoring,” IEEE Transactions on Biomedical Engineering, vol. 64, no. 4, pp. 859–869, 2016.

[2] F. P.-W. Lo, C. X.-T. Li, J. Wang, J. Cheng, and M. Q.-H. Meng, “Continuous systolic and

diastolic blood pressure estimation utilizing long short- term memory network,” in 2017 39th annual international conference of the IEEE engineering in medicine and biology society (EMBC), pp. 1853–1856, IEEE, 2017.

[3] K. R. Vardhan, S. Vedanth, G. Poojah, K. Abhishek, M. N. Kumar, and V. Vijayaraghavan, “Bp-net: Efficient deep learning for continuous arterial blood pressure estimation using photoplethysmogram,” in 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1495–1500, IEEE, 2021.

[4] A. Arjomand, A. Boudesh, F. Bayatmakou, K. B. Kent, and A. Mohammadi, “Transforhythm: A transformer architecture conductive to blood pressure estimation via solo ppg signal capturing,” arXiv preprint arXiv:2404.15352, 2024.

[5] A. E. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, T. J. Pollard, S. Hao, B. Moody, B. Gow, et al., “Mimic-iv, a freely accessible electronic health record dataset,” Scientific data, vol. 10, no. 1, p. 1, 2023.

[6] C. Wuerich, F. Wichum, O. El-Kadri, K. Ghantawi, N. Grewal, C. Wiede, and K. Seidl, “Blood pressure estimation based on electrocardiograms,” Current Directions in Biomedical Engineering, vol. 8, no. 2, pp. 53–56, 2022.

[7] K. M. Syah, K. N. Pramudia, and M. Rochmad, “Implementation of artificial neural network for real-time blood pressure estimation using ecg signal,” in 2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), pp. 12–18, 2023.

[8] M. Simjanoska, M. Gjoreski, M. Gams, and A. Madevska Bogdanova, “Non-invasive blood pressure estimation from ecg using machine learning techniques,” Sensors, vol. 18, no. 4, 2018.

[9] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, “Mimic-iii, a freely accessible critical care database,” Scientific data, vol. 3, no. 1, pp. 1–9, 2016.

[10] Gonzalez-Landaeta, R., Ramirez, B., & Mejia, J. (2022). Estimation of systolic blood pressure by Random Forest using heart sounds and a ballistocardiogram. Scientific Reports, 12(1), 17196.

[11] Azad, R., Asadi-Aghbolaghi, M., Fathy, M., & Escalera, S. (2019). Bi-directional ConvLSTM U-Net with densley connected convolutions. In Proceedings of the IEEE/CVF international conference on computer vision workshops (pp. 406-415).

[12] A. Esmaili, M. Kachuee and M. Shabany, "Nonlinear Cuffless Blood Pressure Estimation of Healthy Subjects Using Pulse Transit Time and Arrival Time," in IEEE Transactions on Instrumentation and Measurement, vol. 66, no. 12, pp. 3299-3308.

[13] S. Baek, J. Jang, S. -H. Cho, J. M. Choi and S. Yoon, "Blood Pressure Prediction by a Smartphone Sensor using Fully Convolutional Networks," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 2020, pp. 188-191.

[14] Elizabeth Gomes, Catherine Liao, Oliver Shay, Nikhil Bikhchandani, April 17, 2021, "A Dataset of Synchronized Signals from Wearable Cardiovascular Monitoring Sensors", IEEE Dataport, doi: https://dx.doi.org/10.21227/3yte-wz05.

[15] Huang, S., Jafari, R., & Mortazavi, B. J. (2023). ArterialNet: Arterial blood pressure reconstruction. 2023 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI). https://doi.org/10.1109/BHI58575.2023.10313518

[16] A. E. Dastjerdi, M. Kachuee, and M. Shabany, "Non-invasive blood pressure estimation using phonocardiogram," in *Proc. 2017 IEEE Int. Symp. Circuits Syst. (ISCAS)*, Baltimore, MD, USA, 2017, pp. 1–4, doi: 10.1109/ISCAS.2017.8050240.

[17] K. Kokkhunthod *et al*., "Blood Pressure Estimation with Phonocardiogram on CNN-Based Approach," *Computers, Materials & Continua*, vol. 79, no. 2, pp. 1775–1794, 2024, doi: 10.32604/cmc.2024.049276.

[18] R. Gonzalez-Landaeta, B. Ramirez, and J. Mejia, “Estimation of systolic blood pressure by Random Forest using heart sounds and a ballistocardiogram,” Scientific Reports, vol. 12, no. 1, p. 17196, Oct. 2022, doi: 10.1038/s41598-022-22205-0.