Final Project Report: Non-Invasive Cardiac Output Monitoring via Multi-modal Cardiovascular Signals and Deep Learning

Demet Tangolar Merna Bibars Michael Cho Abdulrhman Ba Nabila Tim Phan
dtangolar3@gatech.edu, merna.bibars@gatech.edu, mcho314@gatech.edu,
abn33@gatech.edu, tphan73@gatech.edu

Abstract

Accurate, continuous monitoring of cardiac output (CO) is crucial for effective resuscitation management in hemorrhagic trauma, yet current gold-standard methods are invasive and impractical in field settings. This study introduces a fully non-invasive and wearable sensing-based approach utilizing electrocardiography (ECG), seismocardiography (SCG), and photoplethysmography (PPG) signals, integrated with deep learning algorithms, to enable CO estimation without requiring baseline calibration. This critical feature makes the model especially suitable for casualty care scenarios where baseline measurements are often unavailable. The proposed methodology is evaluated on a porcine model (n=6) subjected to controlled hemorrhage and resuscitation protocols. Clinically validated cardiovascular features are be used as inputs for DeepConvLSTM models. Additionally, multimodal raw signal forms are used as inputs for the DeepConvLSTM and transformer models to estimate CO. This work highlights the potential for intelligent, non-invasive CO monitoring systems to improve clinical and trauma care outcomes.

1. Introduction

Cardiac output (CO), a key indicator of cardiovascular health, is essential for monitoring a patient's response to hemorrhage and guiding timely clinical intervention. Despite its clinical importance, CO is typically measured through invasive techniques such as catheter-based thermodilution, which are unsuitable for use outside of hospital settings. This limitation is especially problematic in emergency care and trauma scenarios, where rapid, reliable, and non-invasive monitoring could be life-saving.

This project aims to develop a non-invasive method for estimating CO using only wearable sensors that capture electrical, mechanical, and optical signals from the body. Unlike conventional systems that require complex infrastructure or baseline calibration, our approach is designed to work with raw sensor data or extracted physiological features alone. This makes it better suited for field deployment, including in ambulances, military settings, or low-resource environments.

Our core goal is to investigate whether deep learning models can accurately infer CO from multimodal time-series data recorded from ECG, SCG, and PPG sensors. We focus specifically on two architectures: DeepConvLSTM for its strength in modeling temporal dependencies, and Transformers for their capacity to model long-range interactions across modalities. In doing so, we explore how different fusion strategies (early, mid, and late) affect model performance and generalization.

Our goal with this project is not only to estimate CO accurately, but also to assess whether wearable-based deep learning solutions can reach a level of reliability that rivals traditional clinical tools. If successful, such a system could offer a scalable solution for real-time monitoring in both civilian and combat casualty care.

2. Related Works

The current clinical gold standard for CO measurement (i.e., catheter-based thermodilution) is invasive, making it unsuitable for many emergency settings [9]. Recent advances in wearable sensing technologies enable non-invasive continuous monitoring of cardiovascular dynamics. Among non-invasive options, although echocardiography and MRI provide accurate CO measurements, they are costly and unsuitable for continuous monitoring. Moreover, impedance cardiography (ICG) suffers from decreased accuracy in patients with high body mass index (BMI), while photoplethysmography (PPG), similar to ICG, faces challenges such as motion artifacts and limited reliability in critical care settings [13]. In contrast, although electrocardiography (ECG), seismocardiography (SCG), and PPG signals can each be affected by noise, combining them in a multimodal approach with machine learning enables more ro-

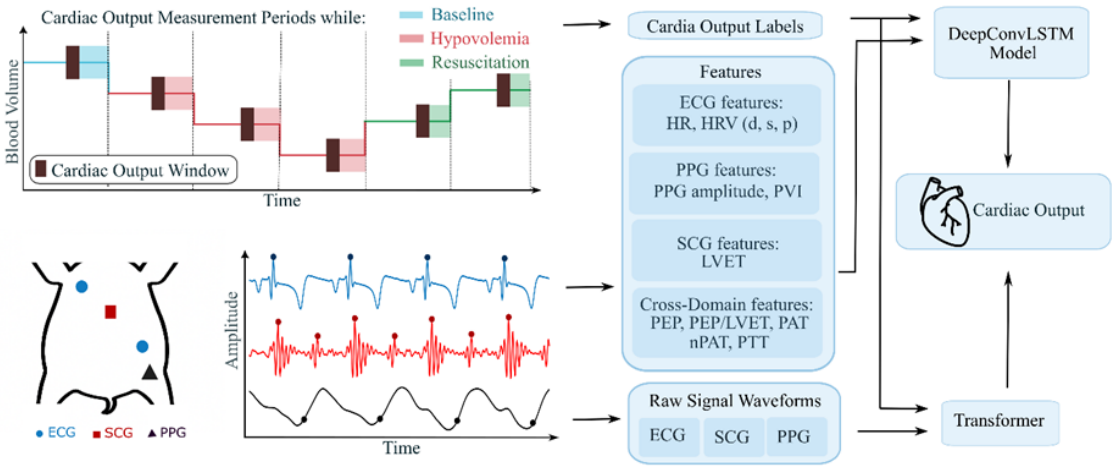

Figure 1. Overview of the methodology: ECG, SCG, and PPG signals were recorded during baseline, hypovolemia, and resuscitation phases, with key features extracted from each modality. These features were used to train a a DeepCOnvLSTM model and raw signals were used to train a transformer to estimate stroke volume, which was combined with heart rate to calculate cardiac output.

bust, non-invasive, continuous estimation of hemodynamic parameters, offering a promising strategy for CO estimation [15].

Several studies have explored this direction. Roha et al. [11] showed that ECG and PPG signals, utilizing 1-D CNN, can indirectly estimate CO for individuals with hypertension by first estimating SV and then multiplying it with HR, outperforming direct CO and BP-based methods. Pastor et al. [9] used PPG signals to classify CO as high or low with 88% accuracy. Ke et al. [5] applied random forest and XGBoost models using arterial pressure waveforms and demographic data, achieving an MSE of 1.42 L/min for CO estimation. Palanques-Tost et al. [7] used ICU datasets and a custom CORE model, reporting a MAPE of 14%. Bikia et al. [1] employed synthetic datasets and ensemble models to estimate aortic pressure and CO with a normalized RMSE of 7.5%. However, the usage of deep learning models is not much explored in the context of CO estimation using multimodal cardiovascular signals.

On the other hand, deep learning models are increasingly more capable of performing multivariate time series analysis. The progress of Recurrent Neural Networks (RNNs) where the recurrent nature of time series data is captured and utilized [12]. Other models that also excel in handling time series data are Long Short Term Memory (LSTM) [3] and Transformer architectures [14] achieving SOTA levels in every time series task and in other domains too. Lin et al utilize an LSTM coupled with an attention mechanism to reduce the error rate on CO predictions [4]. In other medical tasks Panwar et al used Recurrent Convolutional Networks (RCN) to estimate blood pressure from PPG achieving MAE of 0.09 for diastolic BP [8]. To predict the blood glucose levels Rabby et al uses a stacked LSTM attaining predictions with RMSE of 6.45 mg/mL [10]. When used to forecast ICU vital signs transformers outperform the best baseline models by 34 percent in MSE. Deep learning methods gain popularity due to their abilities in performing many multivariate time series tasks, creating a variety of opportunities to improve the current provided baseline of performances.

3. Method

3.1. Model Architectures

We investigate three primary architectural families for SV regression using multi-modal time-series signals: (1) DeepConvLSTM-based models with early/mid/late fusion using cardiac features, (2) Transformer-based models with early/mid/late fusion and (3) CNN-LSTM using wearable raw cardiac waveforms (i.e., ECG, PPG, and SCG). Our goal is to exploit both the temporal and modality-specific dynamics of ECG, SCG, and PPG signals.

We separate our features to represent 4 modalities based on the nature of each signal in medically relevant groupings. The 4 modalities we chose are: Electrical (Heart Rate (HR), HRV Difference, HRV Poincaré, HRV Spectral), Mechanical (LVET), Optical (PPG Amplitude), and Cross Domain (PAT, nPAT, PVI, PEP, PEP/LVET, PTT).

Each row in the original dataset corrosponds to the 12 features recorded at a timestamp. Giving the data a shape of [T, D]. Where T is the number of time stamps, and D is the 12 features. We use a sliding window of size W. The total number of rows is then N = T - W + 1

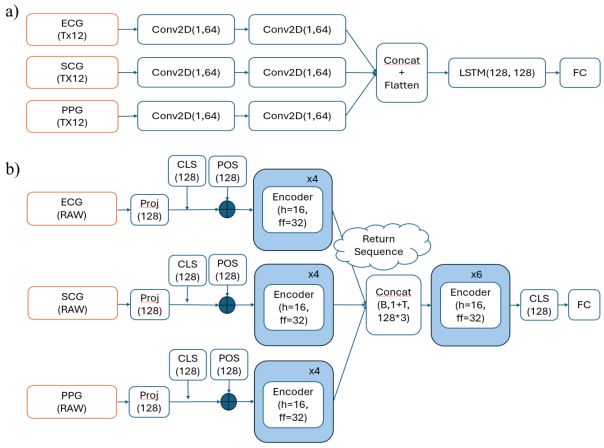

Figure 2. Model Architectures with Mid Fusion a) DeepConvLSTM b) Transformer

3.1.1 DeepConvLSTM Architecture for Features

Inputs: Each modality input has shape $(T \times 12)$, where T is the number of time steps and 12 represents extracted hand-crafted features per window. We used the data from the start of CO measuring time until the end of the protocol step. Ee used a sliding with of 99 beats with stride of 10 for.
Early Fusion: As some modalties have so little of data, we also tried fusing them in the beginning, which means getting rid of the multimodality.
Mid-Fusion: Each modality passes through two Conv2D layers. The resulting embeddings are concatenated and flattened, followed by an LSTM layer. Its output is passed to a fully connected (FC) layer to regress CO.
Late-Fusion: Each modality is processed independently through Conv2D and LSTM layers. The last hidden states are concatenated and passed to an FC layer for regression.
Use of Conv2D on 1D Signals: Although the signals are one-dimensional, we use Conv2D layers with kernel size (1×64) by reshaping data to $(\text{channels}, \text{time})$. This applies convolution along time, maintaining feature/channel structure. This design mirrors Ordóñez and Roggen's architecture [6].

3.1.2 Raw Signal Transformer Architecture

Inputs: Raw ECG, SCG, and PPG signals are projected to 128 dimensions and tokenized using positional and `[CLS]` embeddings, following the Vision Transformer (ViT) structure [2].
Early-Fusion: Each modality is concatenated into a single sequence and processed together by one Transformer encoder, with no separate per-modality branches.
Mid-Fusion: Each modality is encoded independently, then the encoded token sequences are concatenated and passed through another Transformer block:

$$\mathbf{h}_{\text{fused}} = \text{Transformer}^{(6)}(\text{Concat}(\mathbf{z}^{\text{ECG}}, \mathbf{z}^{\text{SCG}}, \mathbf{z}^{\text{PPG}})) \quad (1)$$

$$\hat{y} = \text{FC}(\mathbf{h}_{\text{fused}}^{[\texttt{CLS}]}) \quad (2)$$

Late-Fusion: After modality-specific encoding, only the `[CLS]` tokens are extracted and concatenated:

$$\mathbf{z}_{\text{fused}} = [\mathbf{z}_{[\texttt{CLS}]}^{\text{ECG}} \,||\, \mathbf{z}_{[\texttt{CLS}]}^{\text{SCG}} \,||\, \mathbf{z}_{[\texttt{CLS}]}^{\text{PPG}}] \quad (3)$$

$$\hat{y} = \text{FC}(\mathbf{z}_{\text{fused}}) \quad (4)$$

3.1.3 Raw Signal DeepConvLSTM Architecture

A similar architecture to that used with features (Section 3.1.1) was applied to raw signals, consisting of three main components. First, the CNN feature extractor uses two sets of convolutional layers, each followed by normalization, non-linear activation, and pooling to capture key patterns, reduce signal length, and stabilize learning. Next, a two-layer LSTM models temporal dependencies in the extracted features. Finally, fully connected layers generate the final prediction by integrating information from both the CNN and LSTM.

3.2. Hyperparameter Search

3.2.1 DeepConvLSTM

We performed an extensive random search over both the model architecture and training hyperparameters to identify the best-performing DeepConvLSTM configuration.

As part of this search, we explored different convolutional block designs, including the **ShallowConvBlock**, which consists of a single Conv2D $\rightarrow$ BatchNorm $\rightarrow$ LeakyReLU $\rightarrow$ MaxPool sequence; the **DeepConvBlock**, which stacks two such convolutional blocks; and the **NoPoolConvBlock**, a variation of the deep block without max pooling. The motivation for trying different convolutional depths and block structures was to balance expressive capacity with the risk of overfitting, especially since we used extracted features and our dataset is of medium size.

We also experimented with the fusion strategy across modalities. Specifically, we tested **early**, **mid**, and **late** fusion to evaluate at which representation level the combination of modalities yields the most meaningful and complementary information for stroke volume (SV) estimation.

The hyperparameters explored include batch size, number of epochs, dropout rate, kernel size, LSTM hidden size, number of LSTM layers, convolutional channel depth, model fusion type, and convolutional block type. The full search space is summarized in Table 1.

3.2.2 Transformer

For our Transformer architectures, we set the model dimension to $d_{\text{model}} = 128$, the number of attention heads to $n_{\text{head}} = 8$, and the number of encoder layers to num_layers $= 4$.

We performed an extensive grid search over both architectural and training hyperparameters to identify the best Transformer configuration for stroke-volume regression. First, we explored the fusion strategy (Early, Mid, Late). Doing so, allows us to compare how modality combination affects prediction.

We then tuned batch size, learning rate, epochs, optimizer (Adam vs. SGD), and dropout rate to balance model capacity and overfitting. This helped identify the most effective fusion scheme and hyperparameters for accurate, robust SV estimates. Full hyperparameter details are listed in Table 2.

Table 1. Hyperparameter Search Grid for DeepConvLSTM

Parameter	Values
Batch Size	{16, 32, 64}
Epochs	{50, 100, 150}
Dropout Rate	{0.2, 0.3, 0.5}
Kernel Size	{3, 5}
LSTM Hidden Size	{32, 64, 128}
Number of LSTM Layers	{1, 2}
Conv Channel Depth	{16, 32, 64}
Fusion Strategy	{Early Fusion, Mid Fusion, Late Fusion}
Conv Block Type	{ShallowConv, DeepConk, NoPoolConv}

Table 2. Hyperparameter Search Grid for Raw Signals

Parameter	Values
Batch Size	{64, 128, 256}
Learning Rate	{0.0001, 0.00005, 0.00001}
Epochs	{10, 30, 100}
Optimizer	{Adam, SGD}
Dropout Rate	{0.1, 0.3, 0.5}
Model Type	{DeepConvLSTM, Early Transformer, Mid Transformer, Late Transformer}

3.3. Loss and Evaluation Metrics

Loss Function: We minimize Mean Squared Error (MSE).

Evaluation Metrics: Root Mean Squared Error (RMSE), Coefficient of Determination (R^2), Mean Absolute Percentage Error (MAPE)

3.4. Validation Strategy

We employ leave-one-subject-out (LOSO) cross-validation to evaluate model generalizability across subjects. In each fold, one pig is held out for testing, and the remaining five are used for training. This allows robust estimation of model performance in subject-independent settings, which is essential for clinical deployment.

Table 3. DeepConvLSTM Performance Results

Pig	Config 1			Config 2		
	RMSE	MAPE	R	RMSE	MAPE	R
1	4.96	8.34	0.77	5.53	10.85	0.80
2	12.16	19.62	0.72	10.93	17.73	0.66
3	8.22	12.66	0.69	10.17	15.37	0.52
4	12.01	12.31	0.21	9.82	12.07	0.40
5	9.01	15.29	0.73	9.33	15.83	0.71
6	14.86	23.61	0.55	12.73	21.23	0.44
Median	**10.61**	**13.97**	**0.71**	**10.05**	**15.60**	**0.61**
Mean	**10.21**	**15.30**	**0.61**	**9.75**	**15.52**	**0.59**

4. Data

The dataset used in this study comprises physiological signals collected from six Yorkshire swine (three female, three castrated male; mean age 127.3 ± 13.5 days; mean weight 63.05 ± 5.93 kg), each subjected to a controlled hypovolemia protocol involving staged blood removal until critical decompensation and resusitation afterwards. At the end of each interval, CO was measured via thermodilution using a Swan-Ganz catheter. This material is based on work supported by the Office of Naval Research under Grant N000141812579, N000142212325, and N000142512219. Data were acquired using a non invasive BIOPAC system capturing electrocardiogram (ECG), seismocardiogram (SCG), photoplethysmogram (PPG) signals. Signals were preprocessed with modality-specific bandpass filters (ECG: 0.5–40 Hz, SCG: 1–40 Hz, PPG: 0.5–10 Hz) and segmented beat-by-beat using ECG-derived R-peaks (Pan-Tompkins algorithm). The total number of beat segments available for analysis is 16,613. The filtered ECG, SCG and PPG signals were used for the transformer model. For the feature-based models, twelve validated features were extracted per beat, including heart rate, pre-ejection period (PEP), left ventricular ejection time (LVET), PPG amplitude, pulse transit time (PTT), and several HRV and pulse timing metrics. These were grouped into four modalities—Electrical, Mechanical, Optical, and Cross-domain—based on physiological origin. Each beat is labelled with a SV label (L/beat) which is calculated by dividing the CO (L/min) measurement value by the corresponding beats heart rate (beat/min).

5. Experiments and Results

5.1. Feature based SV estimation

We have implemented Linear, LASSO, Ridge, Random Forest and XGBoost We evaluated the effectiveness of our DeepConvLSTM models for SV estimation by conducting

Table 4. Raw Signals CNN-LSTM vs. Early Fusion Transformer Performance

Pig	CNN-LSTM			Early		
	RMSE	MAPE	R	RMSE	MAPE	R
1	4.69	8.84	0.79	12.22	33.06	0.25
2	7.48	11.17	0.15	8.03	12.12	-0.09
3	11.33	22.77	0.64	13.72	29.95	0.35
4	15.28	18.35	-0.39	19.77	21.53	0.16
5	9.37	12.92	0.12	15.51	20.72	-0.08
6	7.73	17.23	0.71	9.52	19.79	0.34
Median	**8.55**	**15.07**	**0.39**	**12.97**	**21.13**	**0.20**
Mean	**9.31**	**15.21**	**0.34**	**13.13**	**22.86**	**0.16**

Table 5. Mid Fusion Transformer vs. Late Fusion Transformer Performance

Pig	Mid			Late		
	RMSE	MAPE	R	RMSE	MAPE	R
1	9.62	20.42	0.61	8.13	14.71	0.54
2	10.19	12.74	0.15	10.96	13.96	0.064
3	10.89	19.07	0.76	11.83	16.89	0.72
4	8.39	10.18	0.76	9.45	10.87	0.42
5	10.39	13.04	-0.12	15.29	17.57	-0.11
6	11.71	21.72	0.26	13.17	21.53	0.27
Median	**10.29**	**16.05**	**0.44**	**11.39**	**15.80**	**0.35**
Mean	**10.20**	**16.19**	**0.40**	**11.47**	**15.92**	**0.32**

a series of experiments using leave-one-subject-out cross-validation across six pigs. To assess the value of modeling temporal and multimodal dynamics, we compared our deep learning models against several classical regression baselines. These baselines included Linear Regression, LASSO, Ridge Regression, Random Forest, and XGBoost. All models were evaluated using three metrics: Root Mean Squared Error (RMSE, in milliliters), Mean Absolute Percentage Error (MAPE, in %), and Pearson's correlation coefficient (R), which captures the strength of the linear relationship between predicted and true SV values. Among the baseline methods, LASSO regression achieved the best performance overall, with an average RMSE of 8.58 mL and a MAPE of 14.63%. These values serve as strong non-deep-learning baselines, providing context for interpreting the results of our DeepConvLSTM models. The performance of the two best DeepConvLSTM configurations from the random search over 50 different architecture and training configurations is presented in Table 3, both performing significantly better than the clinically acceptable range for non-invasive measurements (i.e., 30%) [9]. One configuration achieved the best MAPE score of 15.30%, using a mid-fusion strategy with NoPoolConvBlock, 32 convolution channels, a dropout rate of 0.2, kernel size of 5, batch size of 128, two LSTM layers, and training for 100 epochs. The best RMSE score of 9.75 mL was achieved by a different configuration: an early-fusion model with NoPoolConvBlock, 16 convolution channels, dropout of 0.3, kernel size of 3, batch size of 128, two LSTM layers, and training for 50 epochs.

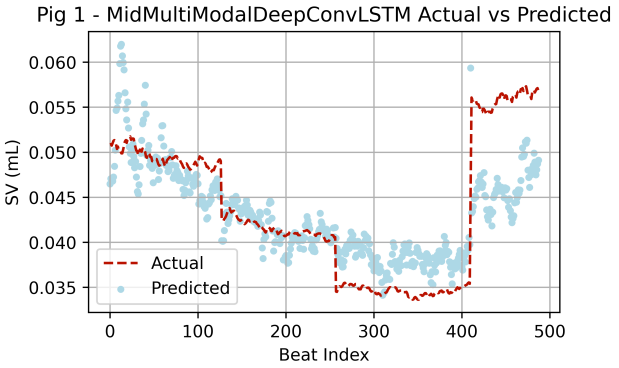

Figure 3. Actual vs Estimated SV Result

These results show that while our DeepConvLSTM models did not outperform the best baseline (LASSO), they were able to achieve competitive performance, particularly in terms of RMSE. The best-performing DeepConvLSTM configurations outperformed Linear, Ridge, and Random Forest regressions, and matched or slightly outperformed XGBoost in RMSE (9.75 vs. 9.53 mL), though lagged behind LASSO in both RMSE and MAPE. This suggests that although temporal modeling provides value, the extracted feature set already contains rich summary information that simpler models like LASSO can effectively leverage.

Fusion strategy notably impacted performance: mid-fusion achieved better MAPE by preserving modality-specific features, while early fusion yielded the lowest RMSE by capturing global patterns early. Removing max pooling (NoPoolConvBlock) also improved generalization, likely by reducing information loss which is crucial for our medium-sized dataset.

In summary, DeepConvLSTM models demonstrated strong generalization and performance competitive with classical baselines. While LASSO remains the most effective model in this setup, DeepConvLSTM offers greater flexibility and potential for scalability to more complex, raw input representations and larger datasets.

5.2. Raw Signal SV Estimation

To evaluate the feasibility of an end-to-end, feature-free approach, we conducted experiments using raw cardiac signals for SV regression. This investigation aimed to determine if deep learning models could effectively learn salient patterns directly from wearable sensor data, bypassing the need for manual feature engineering. We benchmarked two prominent architectures: a hybrid Convolutional Neural

Table 6. SV Estimation Performance Results for Baseline Models

Pig	Linear Regression			LASSO			Ridge Regression			Random Forest			XGBoost		
	RMSE	MAPE	R	RMSE	MAPE	R	RMSE	MAPE	R	RMSE	MAPE	R	RMSE	MAPE	R
1	17.20	33.97	0.89	11.04	22.58	0.90	16.44	32.58	0.89	7.94	13.50	0.83	8.54	13.67	0.90
2	12.64	23.32	0.66	9.72	15.91	0.85	10.62	19.98	0.84	8.01	14.39	0.92	8.30	14.64	0.92
3	18.61	29.84	0.33	9.07	13.99	0.76	18.54	29.68	0.31	5.75	8.18	0.88	8.59	13.12	0.88
4	22.09	31.47	0.87	7.23	9.62	0.92	22.69	32.51	0.87	10.40	12.43	0.88	13.31	16.07	0.85
5	14.68	30.08	0.84	8.98	16.01	0.89	12.40	24.16	0.86	9.45	12.20	0.86	9.18	10.88	0.87
6	4.07	7.17	0.92	5.45	9.68	0.92	4.31	7.31	0.92	16.20	29.74	0.80	9.28	16.80	0.84
Median	15.25	24.48	0.55	**8.65**	**14.31**	**0.79**	14.64	22.95	0.57	10.45	15.81	0.70	9.50	14.38	0.74
Mean	14.88	25.98	0.75	**8.58**	**14.63**	**0.87**	14.17	24.37	0.78	9.63	15.07	0.86	9.53	14.20	0.88

Network-Long Short-Term Memory (CNN-LSTM) model and a Transformer model with varying fusion strategies.

Our experimental setup was designed to ensure robust and generalizable results. The models were trained on raw ECG, PPG, and SCG signal segments. All training and evaluation were conducted using a rigorous leave-one-subject-out (LOSO) cross-validation strategy to assess model performance on unseen subjects. The performance was quantified using three standard regression metrics: Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the Pearson correlation coefficient (R) to measure the linear relationship between predicted and actual SV labels. The computational workload was managed using several NVIDIA H200 GPUs, each equipped with 120GB of VRAM.

The results, summarized in Table 4 and Table 5, reveal a clear performance advantage for the CNN-LSTM architecture. The CNN-LSTM model consistently outperformed the Transformer models across almost all evaluation metrics and subjects. The CNN-LSTM model achieved a mean RMSE of 9.31 mL and a MAPE of 15.21%, outperforming the Early, Mid, and Late Fusion Transformers, which recorded RMSEs of 13.13, 10.2, and 11.47 mL, and MAPEs of 22.86%, 16.19%, and 15.92%, respectively. This superior performance suggests that the hybrid CNN-LSTM architecture is particularly better-suited for this task, as its convolutional layers can effectively extract local, high-frequency features from the signal waveforms, while its LSTM layers model the temporal dependencies between them. In contrast, the Transformer models may have struggled to learn meaningful long-range dependencies from the noisy, complex raw signals without a larger dataset or more specialized pre-training.

Overall, these findings underscore the potential of using raw physiological signals for accurate, non-invasive SV estimation and highlight the effectiveness of hybrid deep learning models for this application.

6. Conclusion

This project presented a non-invasive approach for estimating SV and CO using multimodal physiological signals acquired from wearable sensors. We demonstrated the potential for accurate and calibration-free CO estimation by using cardiac signals.

Our DeepConvLSTM models, optimized through extensive random search and evaluated with LOSO cross-validation, achieved performance competitive with classical machine learning baselines. Notably, the mid-fusion strategy paired with a NoPoolConvBlock yielded the lowest MAPE, while early fusion offered the best RMSE. These findings indicate that both temporal dynamics and modality-specific representations contribute meaningfully to SV estimation.

Similarly, our end-to-end models using raw signals were optimized through the same rigorous random search and LOSO cross-validation methodology. While these models demonstrated promising results, their performance did not match that of the models trained on engineered cardiac features. The CNN-LSTM architecture, in particular, yielded more favorable outcomes than the Transformer archtictures, though its Pearson R correlation remained considerably lower than that achieved by the feature-based approaches. This performance gap suggests several avenues for future work. We plan to explore more complex and robust architectures, potentially increasing model depth. Furthermore, a hybrid approach that combines raw signals with extracted features could be investigated to leverage the strengths of both representations. Finally, transitioning from single-beat analysis to using a sliding window or aggregated beat sections could better capture the broader temporal context, potentially improving estimation accuracy.

Overall, this work lays a strong foundation for intelligent, wearable-based hemodynamic monitoring systems, with direct implications for trauma care, remote health monitoring, and resource-constrained clinical environments.

7. Discussion on Deep Learning Aspects

Our task is formulated as a regression problem to estimate continuous CO from multimodal physiological signals (ECG, SCG, PPG). DeepConvLSTM models were used to capture short-term temporal dynamics in extracted feature

windows, while Transformer architectures handled long-range dependencies across raw signal sequences. Both architectures reflect the sequential and multimodal structure of the data.

Learnable components included convolutional, LSTM, and attention layers, along with final fully connected layers. Non-learned components involved deterministic preprocessing: physiological feature extraction, window segmentation (99 beats, stride 10), and LOSO cross-validation. DeepConvLSTM models used windows shaped as $(T, 12)$, while Transformer models consumed projected raw signals with positional encodings and a `[CLS]` token and both of them outputted a single regressed SV value per input window.

The models used Mean Squared Error (MSE) as the loss function, which is well-suited for regression tasks as it penalizes large deviations between predicted and true CO values. We did not observe overfitting in our DeepConvLSTM or Transformer models. To prevent overfitting, we incorporated dropout layers and used LOSO cross-validation. Training and validation loss curves were monitored throughout, and for the reported configurations, validation losses decreased smoothly along with training loss. In contrast, signs of overfitting, such as decreasing training loss accompanied by increasing validation loss—were not present in our final results.

We performed a random search over key hyperparameters, including convolutional block type, fusion strategy, and training settings. Model performance was evaluated using RMSE, MAPE, and Pearson's R. Notably, the NoPoolConvBlock consistently led to better results, likely due to reduced information loss. Fusion strategy also impacted performance: mid-fusion achieved the best MAPE, while early fusion yielded the lowest RMSE. As for our Transformer models, mid-fusion achieved the best overall results amongst the different fusion strategies, possibly due to its ability to effectively integrate both temporal dynamics and signal interactions at an intermediate representation level. All models were implemented in PyTorch and optimized using Adam.

8. Team Contributions

Individual team member contrubitions can be seen in Table 4.

References

[1] V. Bikia, T. G. Papaioannou, S. Pagoulatou, G. Rovas, E. Oikonomou, G. Siasos, D. Tousoulis, and N. Stergiopulos. Noninvasive estimation of aortic hemodynamics and cardiac contractility using machine learning. *Scientific Reports*, 10(1):15015, 2020. 2

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[3] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 2

[4] H. Kayange, J. Mun, Y. Park, J. Choi, and J. Choi. A hybrid approach to modeling heart rate response for personalized fitness recommendations using wearable data. *Electronics*, 13(19):3888, 2024. 2

[5] L. Ke, A. Elibol, X. Wei, C. Liao, W. Wei, and N. Y. Chong. Machine learning algorithm to predict cardiac output using arterial pressure waveform analysis. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1586–1591, 2022. 2

[6] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1), 2016. 3

[7] E. Palanques-Tost, R. Pallarès-López, R. Padrós-Valls, S. Song, E. Reinertsen, T. W. Churchill, P. Stockwell, E. Pomerantsev, J. Garasic, T. M. Sundt, et al. Cardiac output estimation in the intensive care unit. *JACC: Advances*, 4(5):101663, 2025. 2

[8] M. Panwar, A. Gautam, D. Biswas, and A. Acharyya. Pp-net: A deep learning framework for ppg-based blood pressure and heart rate estimation. *IEEE Sensors Journal*, 20(17):10000–10011, 2020. 2

[9] C. A. Callejas Pastor, C. Oh, B. Hong, and Y. Ku. Machine learning-based cardiac output estimation using photoplethysmography in off-pump coronary artery bypass surgery. *Journal of Clinical Medicine*, 13(23):7145, 2024. 1, 2, 5

[10] M. F. Rabby, Y. Tu, M. I. Hossen, I. Lee, A. S. Maida, and X. Hei. Stacked lstm based deep recurrent neural network with kalman smoothing for blood glucose prediction. *BMC Medical Informatics and Decision Making*, 21:1–15, 2021. 2

[11] V. S. Roha and M. R. Yuce. Direct estimation vs. indirect metrics: Machine learning techniques for cardiac output estimation. In *Proceedings of the 2024 IEEE SENSORS*, pages 1–4, 2024. 2

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. 2

[13] A. Tandon, S. Bhattacharya, A. Morca, O. T. Inan, D. S. Munther, S. D. Ryan, S. Q. Latifi, N. Lu, J. J. Lasa, B. S. Marino, et al. Non-invasive cardiac output monitoring in congenital heart disease. *Current Treatment Options in Pediatrics*, 9(4):247–259, 2023. 1

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017. 2

[15] J. Zia, J. Kimball, C. Rolfes, J.-O. Hahn, and O. T. Inan. Enabling the assessment of trauma-induced hemorrhage via smart wearable systems. *Science Advances*, 6(30):eabb1708, 2020. 2

Student Name	Contributed Aspects	Details
Demet Tangolar	Data Loader, Baseline Models, DeepConvLSTM Implementation	Developed the base data loader for extracted features, implemented baseline machine learning models (Linear, LASSO, Ridge, Random Forest, XGBoost), designed and executed random search over DeepConvLSTM configurations including custom convolutional blocks, report writing and figure generation
Merna Bibars	Data Loader, DeepConvLSTM, Transformer, Training Code Development	Deep Learning models DeepConvLSTM and Transformer implementation. DeepConvLSTM Data Loader implementation for extracted features and their preprocessing. DeepConvLSTM Early, Late and Mid Fusion Experimentation. Implementation of training and cross-validation code for DeepConvLSTM experimentation
Michael Cho	Dataset pre- processing, GitHub management, Signals Dataloader, CNN/CNN-LSTM/Transformer for signals	Dataset cleaning and preprocessing, GitHub management, Created Dataloader for raw signals, Implemented CNN, CNN-LSTM, and Simple Transformer for raw signal/SV estimation,
Abdulrhman Ba Nabila	Initial training pipeline for DeepConvLSTM	Integrated the DeepConvLSTM in the training loop, handled shape mismatches and got the training initial results with no hyperparameter tuninig. Implemented few simple linear classifiers to compare with results from Demet's part to insure consistency with the already established Leave one Out validation
Tim Phan	Mid and Late fusion transformer models for raw signals	Adapted and implemented Mid Fusion and Late Fusion Transformer architectures for SV regression. Integrated models into the training pipeline. Evaluated and compared performance across pigs and different models.

Table 7. Contributions of team members.