POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

Optimization of Deep Neural Networks for PPG-based Blood Pressure Estimation on Edge Devices

Supervisors

Prof. Daniele JAHIER PAGLIARI

Prof. Alessio BURRELLO

Prof. Xiaying WANG

Prof. Luca BENINI

Ph.d. Giovanni POLLO

Candidate

Francesco CARLUCCI

December 2024

Abstract

Continuous and non invasive blood pressure monitoring is crucial for hypertension diagnosis and cardiovascular diseases prevention. Photoplethysmography (PPG) sensors offer a promising solution to solve this challenge, but current estimation methods lack the precision needed to meet medical standards and do diagnosis.

The most promising approach is based on deep learning models that have achieved remarkable accuracy in controlled environments; on the other hand, their deployment on wearable devices faces fundamental constraints. The massive computational requirements and memory footprint of these neural networks make them unfit for edge devices, which must operate within strict power and resource limitations.

Hence, the current challenge resides in maintaining high estimation performance while using more lightweight Deep Neural Network (DNN) models that can fit the constraints of ultra low power edge devices, such as smartwatches.

To cope with this challenge, this thesis proposes a fully automated DNN pipeline encompassing HW-aware Neural Architecture Search (NAS), Pruning and Quantization to generate models deployable on an ultra-low-power multicore System-on-Chip (SoC), GAP8.

This pipeline leverages the gradient-based DNN optimization algorithms available in the PLiNIO library: SuperNet for coarse differential NAS, Pruning-In-Time (PIT) for architecture refinement and Mixed-Precision-Search for Quantization Aware Training.

This thesis go throughout three main contributions for lightweight PPG-based blood-pressure estimations: i) first, we did a preliminary investigation on the relation between the PPG and Arterial Blood Pressure signals and on the impact of commonly used techniques like regularization or data augmentation to adapt the training of the new automatically-searched models;

ii) then, we selected four open-source benchmarking datasets and two "seed" models, i.e., state-of-the-art deep learning models to be used as starting points for our optimization pipeline; in particular, we selected models for two different approaches: either a direct signal-to-label regression or the reconstruction of the whole Arterial Blood Pressure signal from PPG, followed by a peak detection. We utilized the best SoA models: a UNet for signal-to-signal PPG-to-ABP reconstruction and a ResNet for direct systolic and diastolic blood pressure regression.

iii) finally, we applied the full pipeline to these models. The first phase of the pipeline obtained optimized architecture by selecting from different layer alternatives, achieving up to 4.99% lower error or a 73.36% parameter reduction at iso-error. By applying quantization at this stage, we showed that all models found can fit in GAP8 memory without loss in accuracy, while SoA networks are too large to fit the limited 512 kB on-chip memory. During the second step, we further refine the models by using the PIT NAS improving the Pareto front on all datasets and reaching a new accuracy record on the biggest three of them. PIT achieved up to 8.4% lower MAE or a 97.5% parameter reduction at iso-error.

ACKNOWLEDGMENTS

I'm deeply indebted to professors Burrello Alessio and Jahier Pagliari Daniele for their trust, advice and support, and to Pollo Giovanni for his constant help and feedback.

To my family, for the patience and affection expressed.

To my friends, for the mutual esteem and respect that unites us.

Scis quae recta sit linea, quid tibi prodest,
si quid in vita rectum sit ignoras?
Seneca

Table of Contents

Acronyms

PPG
photopletysmograph

ECG
electrocardiogram

ICG
impedance cardiography

MAE
mean absolute error

ME
mean error

SD
Standard Deviation

SoA
state of the art

BP
blood pressure

SBP
systolic blood pressure

DBP
diastolic blood pressure

ABP

arterial blood pressure

PTT

pulse transit time

PAT

pulse arrival time

PEP

pre ejection period

PWV

pulse wave velocity

CVD

cardiovascular diseases

MK

Moens Kortweg equation

AAMI

Advancement of Medical Instrumentation

BHS

British Hypertension Society

ESH

European Society of Hypertension

ISO

International Organization for Standardization

ML

Machine Learning

CNN

convolutional neural network

RNN
recurrent neural network

TCN
temporal convolutional network

DNN
deep neural network

DL
deep learning

NAS
neural architecture search

PIT
pruning in time

MPS
mixed precision search

QAT
quantization aware training

SoC
system-on-chip

DA
Data Augmentation

LSTM
Long Short-Term Memory

ReLU
Rectified Linear Unit

Chapter 1

Introduction

Multiple recent advancements of Machine Learning (ML) throughout industry and academia established it as a general purpose technologies. Similar to mechanization, electronics and automation in previous centuries, machine learning has emerged as a foundational innovation with broad uses across numerous domains.

The wide range of ML applications stems from the flexibility and generality of the underlying techniques. At its core, ML aims to automate the process of learning from data, enabling systems to recognize patterns autonomously for each specific task.

Researchers in the field develop general models and algorithms, such as neural networks and gradient descent which form the backbone of Deep Learning. These methods enable learning from vast amount of data, making ML techniques capable of addressing a wide variety of problems.

Some of the most well-known applications of ML are Natural Language Processing, which encompasses tasks such as understanding and generating text, as well as translating between different languages. Another major domain is Computer Vision, which involves all the tasks where machines need to understand digital images in order to process them or identify objects inside of them. Reinforcement Learning, often utilised in robotic control and game-playing agents, has also seen significant breakthroughs. For example, AlphaZero is a system that has revolutionized game strategy through self-learning and opened new possibilities in complex challenges, like protein folding, showing potential in solving problems that were intractable for traditional methods.

Healthcare is a prominent field where the use of Machine Learning techniques has recently become widespread. Nowadays, deep learning models are employed for tasks such as analysing medical images, design drugs and associate diseases to symptoms. ML techniques proved effective in processing temporal data from sensors, enriching our understanding of them, identifying recurring patterns and enhancing the accuracy and sensitivity of the sensors themselves. Specialized architectures for signal denoising have further improved measurement reliability in noisy environments.

Control of dynamical phenomena or processes requires continuous monitoring of the current state of the system by measuring its important variables. To do so, it is frequently necessary to process sensor data to extract meaningful measurements. This is especially critical in the healthcare field, where the complexity and delicacy of human body makes it difficult to observe internal variables directly. A peculiar challenge of medical sensors lies in their need to assess physiological activities non-invasively - without penetrating the body with instruments or breaking the skin. Furthermore, prompt detection of worsening conditions or emergencies is of paramount importance in any medical contexts. For many clinical applications, continuous, non-invasive and portable tracking of biosignals is essential.

To address the specific demands of this highly relevant field, traditional signal processing techniques are increasingly being integrated with machine learning approaches. This synergy enables the extraction of valuable hidden knowledge from biological signals, enhancing diagnostic capabilities and fostering prevention efforts.

However, modern ML models, particularly neural networks, pose challenges due to their considerable memory footprint – arising from the trained model weights – and substantial energy consumption. This requirements make deploying models on low-power devices challenging. As a result, the prevailing paradigm until now has been to offload data to the cloud where all the processing happens. While effective, this solution introduces significant network usage, latency and privacy issues, especially when processing sensitive biometric data.

Integrating the data processing systems into portable devices, such as wearables, offers great potential for extending prevention to larger populations at an affordable cost. However, edge devices like smartwatches require specifically designed lightweight models that balance accuracy and efficiency within acceptable levels, optimizing the use of limited resources.

This works explores the hurdles of creating deep learning neural networks for biosignal processing on constrained devices, focusing on the specific task of blood pressure estimation from Photoplethysmogram (PPG) signals.

Blood pressure (BP) is a vital sign of utmost importance for cardiovascular diseases management as well as for early diagnosis of asymptomatic hypertension. Continuous and widespread BP monitoring could considerably improve collective quality of life through diffused preventive care.

Traditionally employed methods for BP measurement are either non continuous, like most cuff-based devices, or invasive, like the gold standard technique that requires arterial cannulation to record continuous arterial blood pressure. In contrast, the PPG signal offers a naturally non-invasive alternative with significant potential for continuous blood pressure tracking, even outside of clinical settings, in a portable and automatic manner.

Although studies have shown that PPG contains much of the information needed to reconstruct blood flow within vessels, estimating BP from PPG remains a challenging task. In fact, very few devices have been validated against clinical standards. Various signal processing techniques and machine learning algorithms have already been applied, but the problem remains open.

Similarly to what happened in numerous other fields, deep learning have been shown to outperform classic methods, as larger datasets becomes more and more available.

This work explores the application of automatic Convolutional Neural Network (CNN) generation algorithms to this task with a dual goal:

- Improving accuracy, increasingly bridging the gap with clinical grade instruments.
- Develop new models as lightweight as possible, in terms of both number of parameters, and so memory footprint, and number of operations, that determines latency and energy footprint, trying to meet the tight constraints of low-power devices.

Given the need to explore this trade-off, we employed Neural Architecture Search (NAS) as a general technique for automatic multidimensional optimization. In particular we used two gradient based algorithms (Supernet and Pruning-In-Time) integrated inside the PLiNIO package [1], comparing the new found architectures to all the previous methods in a comprehensive benchmark [2] over 4 different datasets. Other common practices have been tested, like regularization and data augmentation for exploratory data understanding.

We obtained a rich set of Pareto optimal solutions in the complexity vs. accuracy space. The best models have been quantized and deployed on a low-power commercial microcontroller (GAP8).

Chapter 2

Background

2.1 The relevance of blood pressure

Cardiovascular diseases (CVD) are a major burden on societies and the leading cause of death globally. They caused an estimated 17.9 million death in 2019 and 38% of the premature deaths due to noncommunicable diseases [3]. Main causes of CVD deaths are heart attacks and stroke, responsible for more than four fifth of deaths. Identifying those at risk is the first step in prevention of premature deaths.

Hypertension, defined as persistently high blood pressure, is a medical condition that affects a large number of people worldwide, around 1.28 billion adults aged 30-79 years [4]. Hypertension is diagnosed if pressure readings are above 140 mmHg for systolic pressure and above 90 mmHg for diastolic pressure, on two different days. It is estimated that 46% of the affected patients are unaware of having the condition [4]. Hypertension is so severely underdiagnosed because it usually has no warning signs or symptoms, making direct blood pressure measurement the only reliable diagnosis method [5].

On the other hand high blood pressure is a major risk factor of multiple chronic pathologies like stroke, heart failure, atrial fibrillation and dementia. For this reasons it is sometimes referred to as the "silent killer". Early diagnosis of hypertension, before any symptom appears, allows treatment or prevention of all its comorbidities too.

Patients affected by hypertension can adopt a number of different lifestyle changes to minimize the risk of developing chronic conditions. Some recommended preventive measures are: decreasing salt consumption, avoiding smoke and alcohol, practicing regular physical activity. To correctly assess the impact of such habits in the long run periodical or continuous BP tracking is essential.

Blood pressure is one of the four main vital signs routinely monitored by doctors or healthcare professionals, along with body temperature, respiratory rate and heart beat. These 4 signals are primary measures to assess the general physical health of the subject. During triage they can give clues to possible diseases, while in hospitalized patients they must be constantly monitored as they can either show progress toward recovery or early warnings of rapid health deterioration events (fever, cardiac arrest, intensive care unit admission) [6].

Blood pressure values consists of two different measurements, the systolic and diastolic pressures. The first is the maximum peak the pressure reaches inside the artery, during systole, the phase in which the heart contracts, pushing blood to the edges. The diastolic pressure is instead the minimum value of the arterial blood pressure wave, it happens when the heart chambers distend and refill, retracting blood from the veins. The waveform of the arterial blood pressure can be seen in figure 2.1, where SBP and DBP mark the systolic and diastolic points, respectively.

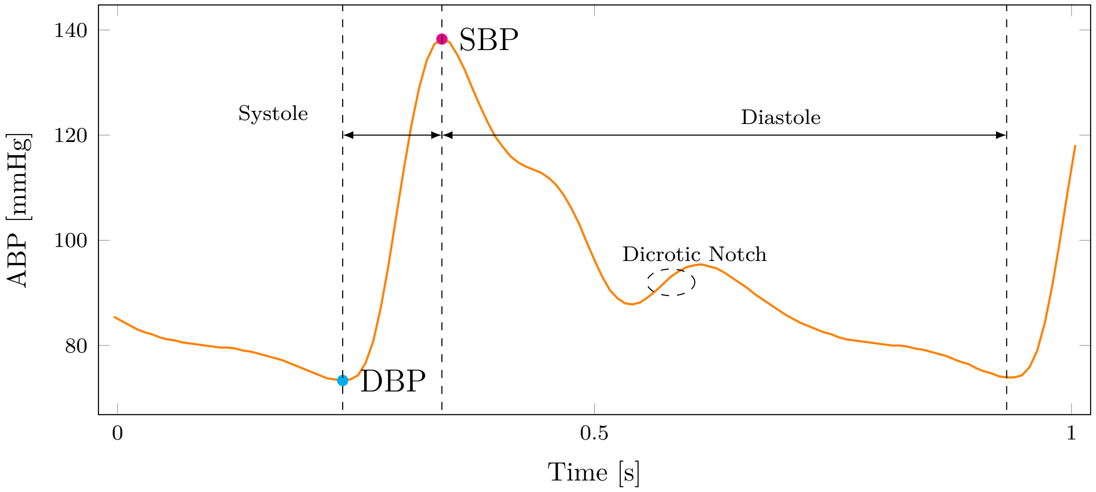


Figure 2.1: An example of an ABP signal obtained through arterial cannulation.

Conventionally, an adult has blood pressure classified as standard if it falls within the range of 100-140 mmHg systolic and 60-90 diastolic.

To control high blood pressure, readings are usually performed during routine clinical visits, where they could be prone to misinterpretation due to masked hypertension or white coat syndrome. The values measured are also discrete and infrequent, failing to give information about how blood pressure changes in a circadian rhythm during the day, during movement or exercise. Moreover ambulatory and nocturnal measurements have been shown to be stronger predictors of cardiovascular risk than diurnal static ones.

2.1.1 Blood pressure measuring methods

The most common devices to measure blood pressure are cuff-based sphygmomanometers. The sphygmomanometer was first invented by Samuel Siegfried Karl von Basch, then perfected by Scipione Riva-Rocci. It is composed of a manometer, either a column of mercury or a digital pressure transducer, and a toroidal rubber bands that goes all around a third of the upper arm. An air bulb and a valve connected to the cuff can pump air inside, applying even pressure to the arm, the entire system and its use are schematized in figure 2.2. The inflatable cuff compresses the artery until complete occlusion, and then the manometer measures the pressure during the controlled release. The minimum and maximum values of pressure, called systolic and diastolic, respectively, are determined listening to Korotkov sounds using a stethoscope, or automatically in digital devices with the oscillometric method that observe pressure fluctuation under the cuff through the transducer.

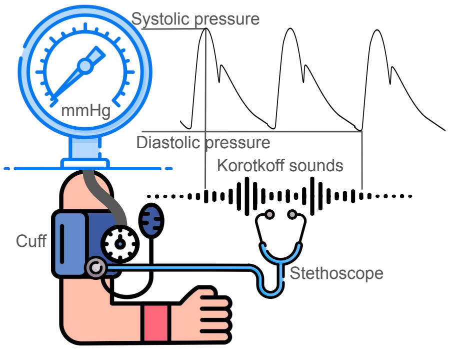


Figure 2.2: A scheme of the sphygmomanometer and its use with a stethoscope through the Korotkoff sounds.

Although this method provides accurate measurements without complex equipment, trained personnel, or invasive procedures, the cuff inflation mechanism prohibits continuous monitoring and makes its use impractical during daily physical activities or any movement. Repeated blocking of blood flow by the cuff could damage tissues, so interval between measurements should be at least 15 - 30 minutes.

Arterial blood pressure can be measured employing an arterial line, a thin catheter with a cannula needle inserted directly inside an artery, usually the radial one, on the wrist. Connecting the cannula to a sterile system with a pressure transducer we can obtain the waveform of the pressure against time. Only used in clinical settings under close supervision of healthcare professionals, this system is accurate and continuous but invasive, exposing the patient to the risk of complications such as infection or bleeding.

2.1.2 Alternative sensors and mathematical models

Ambulatory blood pressure control, computed throughout the full 24 hour sleep-wake cycle, and automatic collection of vital signs through wearables have been attempted using multiple sensors. The most common are the electrocardiogram (ECG) and the photopletysmograph (PPG).

Electrocardiography

ECG measure the electrical activity of the heart for each cardiac cycles. It does so using electrodes placed on the skin that measure the small changes in voltage caused by the depolarization and then the repolarization of the cardiac muscle. Wearables usually integrates a one-lead ECG, measuring heart's electrical activity on the wrist or finger. Single-lead ECG are significantly less accurate than standard 12 or 6 lead devices used in hospitals.

Photopletysmography

Photoplethysmography is the measure (*gràphein*=to write) of a change in volume (*plēthysmòs*=increase) obtained through optical means. The main components of the sensor are a LED light and a photodiode. The LED illuminates the skin, the photodiode measures the amount of light either reflected or transmitted, a scheme is reported in figure 2.3. Every time the heart pumps blood throughout the body a pressure pulse propagates as a wave in the vessels. Although the wave arrives damped to the periphery, it still causes arteries to distend, increasing their volume and thus the absorption of the light. There are no widely accepted measurement units for PPG, they are usually reported as arbitrary units or volts from the diode, in the orders of mV. The amplitudes of the baseline of the PPG signal is often called DC component, for an analogy to the direct current in the electrical domain. The pulsatile component is instead referred to as AC. An example of a PPG signal is showcased in figure 2.4

Pulse Arrival Time

The most common method to assess blood pressure from non-invasive sensors is through the use of Pulse Arrival Time (PAT), defined as the time taken by the arterial pulse to travel from the heart to a peripheral place. The standard is to consider the time between the R-wave peak of the ECG and the detection of the peak at the finger, usually done through PPG. This measurement method is shown in figure 2.5.

It always requires two different sensors, complicating its integration in wearables such as smartwatches or smart rings.

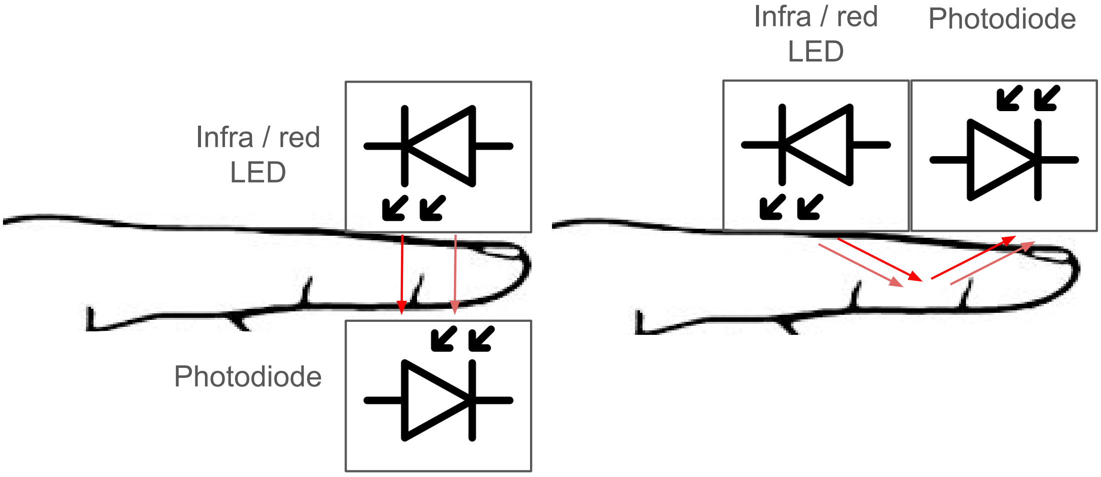


Figure 2.3: A schematic of how transmittance and reflectance based PPG sensors work

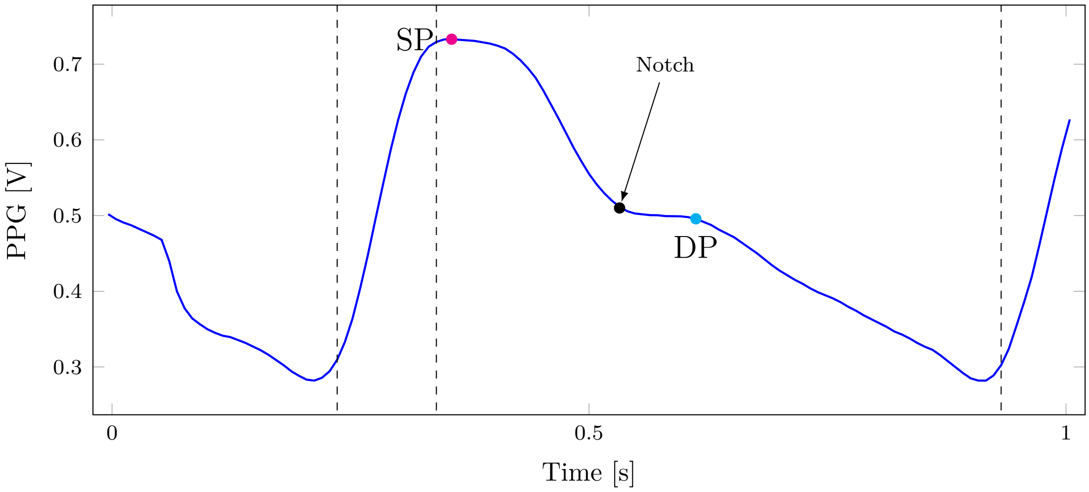


Figure 2.4: An example of a PPG signal

It is possible to evaluate BP from PAT because travel time is related to the speed of the flow, that is, in turn, related to the pressure and the stiffness of the arterial walls.

Pulse Transit Time

BP could also be measured using two PPG signals at different places along an artery, for example, on the upper arm and on the wrist or one after the other on

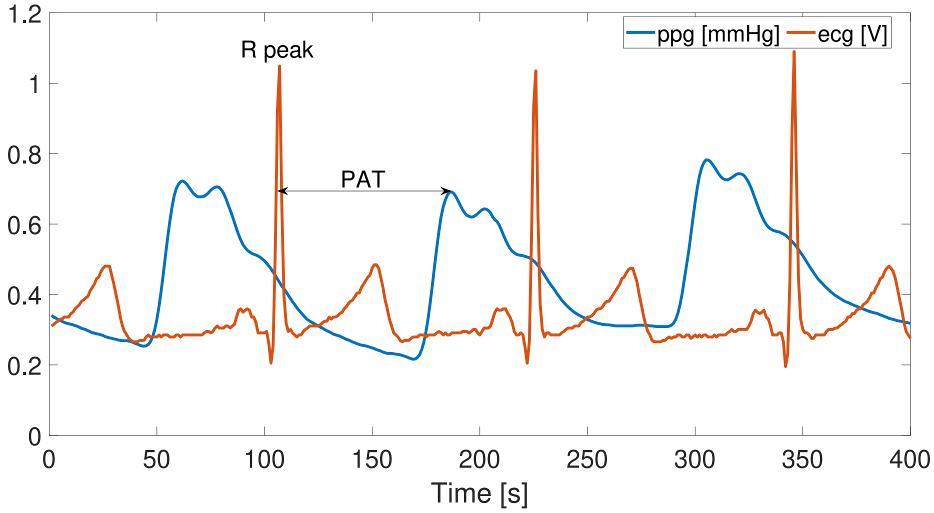


Figure 2.5: The computation of the Pulse Arrival Time from ECG and PPG

the same finger. These are called proximal and distal sites. The time taken for the wave to travel between the two sites is called Pulse Transit Time (PTT). The ECG reacts to the heart electrical processes only, so it measures the electrical impulse to the muscle but not its actual mechanical contraction.

Because of that, the difference between PTT and PAT is in the time between the voltage differences and the exact moment when blood is ejected. This time is called the pre-ejection period (PEP).

The PEP can be non-invasively measured using impedance cardiography (ICG). This sensor estimates how blood volumes change by measuring variations in the electrical impedance of the underlying tissues. Unfortunately, ICG requires multiple electrodes, as well as ECG, usually four placed on the thorax.

Pulse Wave Velocity

PTT or PAT are useful because they allow us to compute the propagation speed of the pulse wave in the artery, the Pulse Wave Velocity (PWV). Then BP estimation becomes a known physical problem: estimating fluid pressure in an elastic tube from flow velocity. The theory of pulse transmission speed dates back to 1808, with the work of Thomas Young. At each contraction of the heart, a pressure wave travels through the arteries, causing them to expand and then contract. The more rigid the arteries are, as happens when the pressure is higher, the faster the wave travels. The artery stiffens as blood pressure increases, increasing PWV accordingly. This relation can be derived from Newton's law applied to a small fluid, where force is the product of density and acceleration.

It is expressed by Frank-Bramwell-Hill equation:

$$PWV = \sqrt{\frac{V \cdot \mathrm{d}P}{\rho \cdot \mathrm{d}V}} \tag{2.1}$$

Where V is volume of the tube per unit of length, P is pressure and ρ is the fluid density. Equation 2.1 holds for an incompressible fluid like blood in an elastic tube like an artery.

An equivalent and more commonly used formulation is the Moens-Korteweg equation:

$$PWV = \sqrt{\frac{E_{\text{inc}}\, h}{2r\rho}} \tag{2.2}$$

where E_{inc} is the incremental elastic module of the vessel wall, i.e. the distensibility, h is the wall thickness and r is the radius of the artery.

To predict the pressure the Moens-Korteweg (MK) equation (2.2) is usually combined with Hughes equation:

$$E = E_0 e^{(\zeta P)} \tag{2.3}$$

where E_0 is the elastic modulus at 0 pressure and ζ is a material coefficient. The MK equation 2.2 equation assumes that the artery wall is a thin shell and that the artery's thickness and radius remain constant as pressure changes, in real applications these hypothesis may not hold. Moreover Hughes equation is purely empirical, so its parameter should be measured on each subject.

Several mathematical models have been developed for fluids in deformable vessels, and many of them have been tested in hemodynamic simulations [7]. Notable advancements are:

- The introduction of an arterial diameter change indicator [8]
- The addition of a viscous flow indicator to account for non newtonian fluid properties of blood
- completely new analytical model that directly correlates BP to PWV without Moens-Korteweg assumptions.

Unfortunately easily interpretable mathematical or statistical models have limited expressive power and reliability. Nevertheless, the complexity of the human body make more sophisticated mathematical modeling of heart-artery systems hard to test and use in practice. A major obstacle is the estimation of internal physiological characteristics, like the distensibility, that widely change between individuals. Using average values for the whole population would make the model too inaccurate for many. Conversely, carrying on extensive black box system identification for every subject in order to produce accurate individualized estimations would have prohibitive costs. Finally, the very need to measure two different waveforms for all PTT methods hinder the practicality and comfort of the whole gear.

2.1.3 Other uses of PPG

PPG is already in use in medical grade devices to accurately estimate blood oxigenation and heart rate. The subsequent section explains how this quantities are estimated in order to highlight some important features of the PPG signal itself. Following paragraphs will then describe why Blood Pressure estimation is a task of a different kind with its own set of challenges.

Pulse Oximetry

Pulse oximetry is the measure of the amount of oxigenated hemoglobin in blood. This can be estimated using PPG through the change in light absorption during the cardiac cycle at different light frequencies, usually red and infrared, according to the Beer-Lambert law. Peripheral oxigen saturation is defined as:

$$S_pO_2 = \frac{HbO_2}{HbO_2 + Hb} \tag{2.4}$$

Where HbO_2 is the oxygenated hemoglobin colored bright red, while Hb is deoxygenated hemoglobin that has a darker hue of red. Their relative concentration determines the color absorption of blood, from which S_pO_2 is inferred:

$$\overline{R} = \frac{\frac{I_{AC}(RED)}{I_{DC}(RED)}}{\frac{I_{AC}(INFRARED)}{I_{DC}(INFRARED)}} \Rightarrow S_pO_2 = 110 - 25\overline{R} \tag{2.5}$$

Where I_{AC} and I_{DC} are the amplitude of the DC and AC components of the PPG signal, respectively.

Monitoring oxygen saturation is particularly important during anesthesia. Additionally, due to the simplicity and inexpensiveness of the setup, pulse oximeters are now also employed in a wide range of clinical settings, including diagnostic purposes of respiratory diseases. They have recently been recommended for home management of COVID-19. However, conventional pulse oximetry still faces some challenges, e.g. values are not reliable during movement and have been found to be less accurate for patients with black skin. Nevertheless, traditional signal processing techniques to filter out noise and clean the PPG are already enough to obtain a clinically useful measure, without any need to resort to machine learning.

Heart rate

The heart rate is another vital sign usually measured with PPG sensors, using the relation between the periodicity of the optical signal and the heart beats. Ideally, the PPG fundamental frequency should correspond to the pulse rate. For this reason inter-beat intervals are computed directly from the time between systolic peaks, often using a sliding window tracking algorithm to account for outliers and make sure estimations don't vary too abruptly. Unluckily, on real PPG signals, especially the one coming from wearables, the accuracy of this simple mathematical methods is degraded by noise and motion artifacts.

Motion artifacts hinder PPG effectiveness disrupting sensor pressure to the skin, causing external light to leak below the wristband and so adding noise to the data.

Many techniques have been tested for denoising, such as peak detection algorithms, model-based adaptive filtering approaches or, more recently, machine learning and deep learning. The latter proved especially effective in reducing the impact of motion artifacts through fusion of inertial sensor data from wearables [9] [10]. Neural architecture search and data augmentation also proved beneficial to this task [11].

2.1.4 PPG characterization

Apart from heart beats the PPG light ray absorption or reflection through the body is influenced by multiple other body movements. As a result, PPG gathers information about the cardiac, vascular, respiratory and autonomic nervous systems. This means that it could be used to monitor numerous physiological phenomena, for example breathing rate, arterial stiffness or atrial fibrillation.

The latter is already available on some commercial devices [12]. If the algorithms developed for these estimations could be refined and validated against medical standards, all wearables could become powerful tools for health enhancement of the many, even informing clinical decisions.

Some of the applications that are being investigated and could revolutionize home medicine in the near future are Obstructive Sleep Apnea detection, mapping the spread of infectious diseases (with the purpose of quickly informing policies of healthcare institutes for better control and spread minimization), sleep monitoring, vascular age assessment, cardiovascular risk prediction or heart failure predisposition, among others.

Breathing influences PPG at subcardiac frequencies, modifying frequency and most of all the baseline wander. Common PPG preprocessing techniques includes filtering out low or high frequency content or removing the baseline wander by fitting polynomials to the points of minima. The exact method used for baseline wander removal, often integrated in the electronics of the data collecting device, can dramatically alter the final wave and should be taken into account during model evaluation.

PPG baseline level, the so called DC component, have been thoroughly studied by physiologist, and can be perturbed by temperature, metabolism, drugs or autoregolatory response of the organism for homeostasis.

Finally venous blood volume also fluctuates and slightly influence the signal.

The shape of the PPG generally features two peaks, the higher one is called systolic while the other diastolic. The valley between them is the dicrotic notch, as shown in figure 2.4 The PPG waveform changes with age, for example the second peak and the notch are usually less visible in elder patients or after exercise. It can also drastically variate based on the measuring site and between individuals, as artery distension depends on elasticity on the internal tissues, strength of the heart and blood viscosity.

2.1.5 PPG modelization

The PPG waveform reflects the movement of the pressure waves in arteries that are closed cavities. For this reason it is determined not only by the incident wave from the heart but also by all the reflected waves from the periphery of multiple vessels [13].

Blood flow velocity affects PPG too, because of reorientation of reflective erythrocytes that may act as mirrors. This effect has been confirmed using a rigid glass pipe, where moving blood creates an oscillating PPG. Nonetheless, it seems negligible compared to absorption of light due to volume increase and so unimportant in-vivo.

Not all the mechanisms underlying the PPG shape are yet fully understood. The large number of components contributing to PPG make it a signal of great potential but also makes it hard to extract single meaningful quantities from it.

In an effort to understand how cardiovascular properties influence the resulting PPG wave multiple physical models have been designed and tested in experiments or simulations, both computational and mechanics [13] [14] [15].

Several properties have been confirmed to affect the wave, especially in the portion between peaks. Among them the diameter of the artery or the peripheral vascular resistance.

Furthermore, the same individual feature is often influenced by multiple internal parameters. Specifically, in our case, general models designed to predict blood pressure should be robust against changes in arterial stiffness and peripheral compliance [16], because all of them may influence the signal in a similar way. That makes model generalization harder and requires to tune the model to each single subject in order to obtain an acceptable accuracy.

2.1.6 PPG *vs* ABP relation

As previously mentioned the amount of absorbed light is directly linked to the arterial blood volume that fluctuates at each heartbeat. On the other hand the relation between blood pressure and non-invasively measurable properties like volume is not so straightforward.

That's why, in spite of the striking resemblance of the PPG signal and the waveform of the Arterial Blood Pressure invasively measured, reconstructing one signal from the other it's a non trivial task.

Considering the relation between ABP and arterial non-optical plethysmography, obtained through direct measures of the diameter of the artery, the two share a similar shape [17]. A complete study of the relation between vascular transmural pressure and arterial volume can be found in [18]. They show how plotting ABP against PPG, the resulting graph has a sigmoid shape, meaning that volume become ever less compliant the higher blood pressure becomes. The AC component of PPG plotted against transmural pressure yields a similar curve. Plotting PPG against ABP we obtain loops instead of curves, proving the presence of hysteresis. Arteries become stiffer as pressure changes more rapidly. This phenomenon, known as dynamic compliance, along with stress relaxation, are intrinsic properties of blood vessels.

This characteristic of the arteries may cause PPG to lack high frequency waveform features of ABP. A vessel also takes some time to relax after mechanical stimuli, so the PPG is not an immediate photograph of internal pressure variations.

2.1.7 Data-driven approaches

Following the recent advancements in data manipulations and signal processing, data-driven solutions to many tasks are on the rise. Machine learning approaches are totally hypothesis-free, compared to any other model-based solution, and they do not require pulse wave analysis algorithms that may be hard to design, because of the peculiarities of each different sensor or individual.

All machine learning models assume that the input contains enough information about the desired output, so it's possible to automatically learn a meaningful relationship. In the case of two different signals, one measured and one to be reconstructed, a strong correlation between them indicates the existence of such relationship.

The most promising signal for blood pressure is PPG. It has a distinctive similarity to Arterial Blood Pressure waveform and correlation studies show it contains most of the information needed to extract ABP [19]. Considering both everyday use feasibility and potential accuracy, in this work only models that receives as input a single PPG signal will be considered.

Deep learning

As bigger datasets become available, deep learning and neural networks have outperformed by far traditional machine learning methods in multiple fields. This is expected to happen for health signals too, in particular for blood pressure. PPG could be fed to our models in three main ways:

- Extracting specific features and fiducial points. This approach is typically adopted for traditional machine learning models. Although features could improve model explainability, they must be manually defined, a laborious task. Moreover, for the considered task, domain knowledge doesn't help in finding meaningful fiducial points, forcing researchers to resort to expensive automatic features selection algorithms.
- Directly feeding a signal sample. This method suits modern convolutional neural networks, that automatically extract embeddings on their own. Resulting model is completely black box but promising performance-wise.
- As a spectogram, after Fourier or Wavelet transform computation. This input format makes techniques and architectures from the image manipulation domain applicable to this purely temporal task. That allows data scientist to leverage the conspicuous knowledge already been accumulated in the computer vision domain.

Problem formulation

From machine learning practitioners point of view the problem could be either a regression or signal translation problem. In the first case given a signal sample or the features extracted from it, the model has to predict two discrete values, the systolic and diastolic blood pressure (SBP and DBP respectively). This approach is referred to as Sig2Lab or Feat2Lab.

In signal translation task instead, the continuous ABP signal is used as label during training and the model needs to reconstruct it from the corresponding PPG sample. In this case the model must be generative and architectures are usually composed of an encoder, that produces embeddings, and a decoder, that generate the target signal. This approach is referred to Sig2Sig.

This work considers the second input format, not the spectogram because of time limits, and the Sig2Sig and Sig2Lab training methods, not the Feat2Lab as it will focus exclusively on neural networks.

Chapter 3

Related Works

3.1 Datasets

Several datasets including the PPG signal are publicly available, varying widely in number of patients involved and samples collected. For our task some misurations of blood pressure is also required, it can be scalar systolic and diastolic values or the continuous Arterial Blood Pressure signal. The largest source of PPG data for number of patients is the UK Biobank project [20] with PPG measurements from 205357 subjects. It includes scalar values of Blood Pressure recorded in the same visit but not simultaneously.

The second biggest project regarding biosignals and health-related data is MIMIC (Medical Information Mart for Intensive Care) [21]. MIMIC is an ongoing project including 4 different generations, each one larger and richer in data. MIMIC-II [22] and MIMIC-III [23] are large database containing information related to patients admitted to intensive care units in a large tertiary hospital. Among the data included there are the most important vital-signs, informations about all exams performed, diagnosis and treatments received. The most recent MIMIC-IV includes also free-text medical notes from the Beth Israel Deaconess Medical Center. All these data warehouses are sourced from databases of electronic health records designed for the everyday work in the hospital. Because of the large mount of sensitive health records all data have been de-identified. The datasets UCI and SENSORS, that will be used in this work are composed of data sourced from MIMIC-III and MIMIC-II. Another valuable resource is VitalDB [24], providing 486451 waveforms including PPG, ECG and BP signals from 6153 patients. In this case the PPG signals have been measured with finger clamps during operations. Although smaller than the previous one, MESA (Multi-Ethnic Study of Atherosclerosis) is still a huge project in which 6814 black, white, hispanic, and Chinese-American men and women took part. PPG was recorded from 2056 of them, while they were undergoing polysomnography.

It's important to emphasize how all this datasets were collected in clinical environments on patients undergoing treatment for some medical conditions. Factors such as drugs assumed, treatments received and the fragile state of the body at the time (especially in intensive care patients) significantly influence the blood pulse. Consequently, blood pressure estimation models trained on photoplethysmograph signals from the currently available datasets may not generalize correctly to healthy populations [25].

3.2 Related Machine learning works

Many studies have already attempted to perform blood pressure estimation fusing PPG to other signals, mainly ECG. The more straightforward approach is to use recurrent neural networks (RNN) architectures trained on ECG and PPG features to predict scalar SBP and DBP values [26].

Another set of widely tested architectures are encoder-decoder based. They are usually trained to reconstruct the ABP signal, forcing the encoder to generate a latent space containing the most relevant PPG features. Then the decoder is discarded and a regressor is trained on the encoder embeddings to predict systolic and diastolic pressures.

In particular in [27] shallow U-Net architectures have been used to predict BP from both PPG and ECG showing good performances on large datasets, like MIMIC II. The model receives as input ECG, PPG and the first and second derivatives of PPG.

Apart from recurrent neural networks, specifically made for temporal data, and autoencoder based network, used to reconstruct, denoise or translate signals, Convolutional Neural Networks can be effective too. Hybrid CNN-LSTM network have been employed in the work [28]. They evaluated model performances through a correlation study on different datasets: MIMIC-II, UKM and PPGBP, claiming that their model works reliably in specific pressure ranges.

A particular kind of pure CNN architectures, called Temporal Convolutional Networks (TCN) can also manage temporal data successfully, outperforming RNN on many tasks related to time series forecast or trajectory prediction. A basic component of TCN are causal convolution layers, a modified convolution where the output at each instant depends only on past timesteps, so the model does not violate the order of the input data. TCN integrates them with all the usual components of CNN, applying all the knowledge gathered from the countless CNN applications to the time series field.

Another approach is trying to improve the PTT method using a machine learning regressor to identify the underlying physical model, bridging the gap from PTT to blood pressure values [29]. First of all PTT is measured using two consecutive PPG sensors placed on the finger, then the PWV is computed knowing the distance between them. Finally the PWV is fed to a Gaussian process regressor that predict blood pressure minimum and maximum values.

An evident drawback of this approach is the need of two separate sensors on the finger, making it cumbersome to integrate into a single wearable device. Moreover, the distance between the sensors should remain constant at all time, so the patient cannot use the hand while measuring. It could still be useful in a clinical settings if its accuracy will rise to match the invasive arterial cannulation techniques it ought to replace.

Predicting Blood Pressure from a single PPG waveform is the approach that holds the most potential for widespread everyday applications, it is therefore the subject of many studies.

Chowdhury et al. [30] combines PPG with demographic informations like age, sex, height, weight and body mass index. These demographic data are fed to a machine learning regressor along with features obtained from the preprocessed PPG waveform, its first and second derivatives. They extracted 107 features from the time domain, the frequency domain and statistical metrics, then tested three different feature selection algorithms on them: ReliefF, CFS and fscmrmr. On the selected features five different machine learning algorithms were trained, of which the best two underwent hyperparameter optimization. Their best model is a Gaussian Process Regressor trained on ReliefF selected features, obtained a 3.02 MAE on SBP and 1.74 MAE on DBP. These metrics would meet the Association for the Advancement of Medical Instrumentation (AAMI) and British Hypertension Society (BHS) standards. The huge number of inputs display the difficulties of dealing with handcrafted features, which is the primary shortcoming of traditional machine learning studies.

In the PPG2ABP study [31] a two stage deep learning method is proposed to directly translate PPG to ABP signals. The first component is called approximation network and consists of a one-dimensional UNet. It creates an approximated version of the ABP. Then a refinement network improve the reconstruction. The second network is a U-Net architecture with residual connections inside the basic building blocks. The whole system achieved a MAE of $3{,}449 \pm 6{,}147$ mmHg for DBP and $5{,}727 \pm 9{,}162$ mmHg for SBP. Unfortunately the total number of parameters or operations is not reported.

The use of two concatenated UNet results in an architecture arguably similar to a WaveNet, already used in many works to translate PPG to ABP [32]. Paviglianiti et al. [33] used PPG, both alone and coupled with ECG, as input to ResNet, WaveNet and LSTM models on MIMIC data and on a custom dataset. The best performing configuration has been ResNet followed by 3 LSTM layers, it obtained 4,118 mmHg and 2,228 mmHg MAE on systolic and diastolic BP respectively.

Recently, transformer based architectures surpassed many previous approaches in Natural Language Processing and Computer Vision tasks, because of their temporal feature representation skills. As a consequence various studies have applied them to time series prediction and regression, including our task. Pure transformer architectures have showed good results on multi-task prediction, both oxygen saturation and blood pressure, using fully connected layers to project the embeddings generated by the transformer to each task. In [34] the authors declare a MAE of $2{,}52 \pm 2{,}63$ and $1{,}37 \pm 1{,}89$ for SBP and DBP respectively.

Other proposals are hybrid architectures including Gated Recurrent Units, Multilayer Perceptrons and attention [35], or convolutional layers and attention [36]. In the latter the input is a multi-channel PPG signal combined with a measure of the pressure of the finger against the sensor. Attention is used to determine the most relevant PPG channel, for each different level of pressure detected. The aim is to make the system more robust to errors caused by different finger positions.

3.3 Commercial products validation

Some cuffless blood pressure monitoring products are already on the market, available either as physical devices, such as ready-to-use wearables, or as mobile applications utilizing smartphone sensors.

The Aktiia bracelet is a wearable device made specifically for blood pressure monitoring. It has been clinically validated in a study [37] on 91 patients, including hypotensive to hypertensive pressure levels. The metrics used are mean error and standard deviation, considering different motionless positions. The device met criterion 1 and 2 of ISO81060-2 standard in the position recommended for sphygmomanometers: sitting with the arm on a desk and wrist at heart level. The study proved the usefulness of such devices for home monitoring, even if they are still slightly less accurate than clinical instruments. The overall performances, however, have some strong limitations. Signal acceptance rate drops significantly even with the slightest motion artifact and the device automatically aborts a measurement when moving. The bracelet also needs to be calibrated each month with a provided automatic cuff based device. Another drawback is the use of closed-source algorithms. Their updates complicates comparative studies, as the same product might use a different estimation system over time. Furthermore, a smartphone is needed to communicate estimated BP data to the user.

The Samsung Galaxy Watch 4 smartwatch provides BP estimation from PPG, and this specific feature underwent validation too [38]. The study involved 20 young patients including darker skin tones and compared the smartwatch against a sphygmomanometer and two automatic oscillometric device, on the arm and on the wrist. They documented a strong SBP correlation and a moderate DBP correlation, but highlighted the same limitations of Aktiia and the considerable cost of the device.

OptiBP [39] is a mobile application that measure BP on the smartphone. PPG is collected placing the user's fingertip on the smartphone camera. On 353 recordings from 91 subjects OptiBP fulfilled validation requirements of AAMI/ ESH/ ISO universal standards. It is worth noting that the app allows for independent non-invasive measurements at any time, but it is not suitable for continuous monitoring.

Seismo is a similar smartphone app that uses both camera's PPG and smartphone accelerometer to detect the opening of the aortic valve, through vibration. This technique is also known as seismocardiography. It computes the PTT from the two collected signals, then applies Moens-Korteweg equation. Among its main limitations are the need for periodic subject-specific calibration and the limited number of patients on which it has been validated [40].

3.3.1 Evaluation and comparison issues

Multiple reviews of previous works points out several issues that are common across many studies [41]. First of all, the variety of employed datasets, many of them custom made and not publicly released, make the declared results not comparable. These datasets tend to be too small to really evaluate deep learning methods, or have a too narrow range of subjects, so the model may not generalize when deployed in production.

Many datasets are also entirely collected in a clinical environment, on patients affected by various ailments. Even if the conditions do not concern the heart directly, they may still influence the PPG readings, making them less representative of the sane population.

Bigger datasets usually share a common raw data source, but every researcher used different preprocessing steps that may heavily affect the final results. Finally, different evaluation metrics or correlation indices are used between different studies, so we can't really rank all the models or evaluate them against standards for clinical instruments.

Another important and often overlooked issue is the incorrect split between training and validation sets. The practice of simply splitting all the data randomly between training, validation and testing, or into different folds for cross validation with uniform probability, leads to subsets with skewed blood pressure distributions and subject information leaking. In the first case hypertensive or hypotensive patients would not be adequately represented during training. In the second case samples of the same patient are in both training and test, leading to over-optimistic results with respect to the real use. In real products a model has to work on data from patients it has never seen before.

3.4 The benchmark

To assess all this problems of fair evaluation, comparison and generalization our work is based on a comprehensive benchmark [2].

3.4.1 The datasets

The benchmark includes 4 standard and publicly available dataset, that will be referred to as SENSORS [42], UCI [43], BCG [44] and PPGBP [45]. All the datasets have a large variety of samples for each subject and wide BP distributions, some statistics about each of them are reported in table 3.1.

Dataset	Amount	Demography (%Male & Age)	Segment Length (s)	Validation Strategy
PPGBP	subjects: 218 segments: 619 duration: <1 hour	46.9% 56.9 ± 15.8	2,1	5-fold CV
BCG	subjects: 40 segments: 3063 duration: ~4 hours	44.5% 34.2 ± 14.5	5	5-fold CV
SENSORS	subjects: 1195 segments: 11102 duration: ~15 hours	59.8% 57.1 ± 14.2	5	5-fold CV
UCI	subjects: unknown segments: 410596 duration: ~570 hours	unknown	5	Hold-One-Out

Table 3.1: Summary of statistics about the four datasets used

SENSORS dataset [35] [42] is a subset of the MIMIC-III database. It involved 1195 patients from intensive care units, including their demographic data as well as both PPG and ABP signals. It's the second biggest dataset in the benchmark after UCI, with a total measurement duration of around 15 hours, and has a medium ratio of segments per patient.

UCI dataset [43] [46], also called Cuff-Less Blood Pressure Estimation Dataset is a subset of MIMIC-II waveform dataset. Although MIMIC-II and MIMIC-III are sourced from the same sets of measurements, made in the same conditions with the same devices, UCI and SENSORS datasets are disjunct subsets. UCI is the biggest dataset we've used, it includes PPG and ABP waveforms but no information about subjects. For this reason data has been split maintaining BP distributions, but there is no way to check for subject information leaking. In spite of that, its size alone should be enough to warrant a decent grade of model generalization.

BCG [44] [47] dataset is a ballistocardiography dataset collected in hospital beds from 40 patients, four of which had some heart conditions while all the others were healthy. They used the Finapres Medical Systems Finometer PRO to measure continuous brachial blood pressure and GE Datex CardioCap 5 for the PPG. The benchmark published a resampled version of the data, from 1000 Hz to 125 Hz, they also rescaled the signal by a factor of 100 mmHg/Volt. BCG is a dataset smaller than the previous ones, comprising around 4 hours of measurements totally, with a consequently high ratio of segments per subject, around 76. Therefore we expect less data variation and a narrower BP distribution.

PPGBP dataset [45] [48] is the smallest one, adding up to less than an hour of readings totally, but involving 219 subjects with different cardiovascular diseases, like hypertension or diabetes. Blood pressure has been measured using Omron HEM-7201 device, so only SBP and DBP discrete values are available. The measuring protocol adopted for this dataset dictate 10 minutes of rest followed by three PPG measurements, each 2.1 seconds long, made with SEP9AF-2 device, these signals have been resampled from 1000Hz to 125Hz too. The really low segments per subject ratio of 3 makes it a dataset with a large data variation for its size.

The datasets PPGBP, SENSORS and BCG are divided in five different folds and all models are trained on them using 5-fold cross validation. For UCI Hold-One-Out policy is adopted instead, considering its size and the lack of subject identification numbers. The split is completely compliant with the adopted benchmark guidelines for fair evaluation. In particular the authors divided data considering subjects and applied a stratified partitioning procedure to avoid cases of underrepresented BP labels.

All the collected data from each dataset underwent the same preprocessing pipeline, ensuring representative data distributions and no information leakage between subjects. All the preprocessing code and the preprocessed datasets have been made public [49] [50].

Finally, 11 different models, from previous works, have been retrained and tested on the same data with common evaluation metrics, warranting comparable and reproducible results. The models already tested were among the best performing up to the benchmark release date and they cover many different approaches to the problem, from traditional machine learning to deep architectures.

The evaluated algorithms utilise all the three different input-output formats:

- **Feat2Lab**, going from handcrafted and selected features to SBP and DBP values. Used by traditional ML models, like support vector machines or random forest.
- **Sig2Lab** goes directly from the whole PPG sample to blood pressure discrete values. Used by some DL networks, like ResNet or SpectroResNet, a ResNet-GRU model.
- **Sig2Sig** models generate a continuous ABP waveform from a raw PPG signal. Used by U-Net, V-Net and PPG2ABP models. These models can't be applied to PPGBP data because there is no continuous ABP waveform to reconstruct.

The first two approaches frame the task at hand as a regression problem, while the third is suited for encoder-decoder architectures that draws experiences from the field of sequence-to-sequence generation or signal denoising.

All the new models generated in this study have been trained and evaluated on the data from the benchmark and have been compared to all the previous SoA models. This work takes into account also model footprint, expressed in terms of memory and operations, something that had never been considered in the benchmark paper or in any other work before.

The benchmark uses Mean Absolute Error (MAE), mean and standard deviation (glsME±SD) to compare all models. They also introduce a new metric, Mean Absolute Scaled Error, defined as a percentage. It is the ratio between the evaluated model's MAE and the MAE of a naive model, the one that always predict the mean value of the training set labels.

This work mainly focuses on MAE to account for performances, adding number of parameters and operations into consideration. The multidimensional comparison is expressed in graphs that will visualize all models and properly map the search space of the NAS algorithms applied.

As expected, among state of the art models in the benchmark, the deep learning networks rise to better performances, competing with traditional machine learning methods, on the biggest datasets. Traditional machine learning methods using the Feat2Lab approach achieve the best performances on BCG and PPGBP, showing there results slightly better than or comparable to the other networks. On the SENSORS dataset the best models are support vector machines and U-Net, while on UCI ResNet and U-Net are undisputed for SBP and DBP, respectively. We can therefore assert that DL methods could undeniably outperform classical data analysis and machine learning techniques, they only need large enough datasets. For this reason this work focused on improving deep learning architectures only.

The metrics reported in the benchmark are noticeably higher than those in most previous works. This is due to the benchmark preprocessing and data splitting techniques previously mentioned, that ensure correct model assessment and are strictly required to meet clinical standards. The high errors obtained mirror reality, where no system operates with sufficient accuracy without an initial personal calibration, and periodical recalibration throughout its use.

Subject-specific calibration is an open problem too. Many different studies have applied various techniques, from building models that are only subject-specific [32] to retraining or finetuning a pretrained model on new data from the target subject. For encoder-decoder architectures, pretraining on large datasets for PPG signal reconstruction can generate meaningful embeddings that should improve performances on the final task of ABP generation or discrete blood pressure values prediction. Domain adversarial neural networks have also been employed to generate personalized models [51].

3.5 Neural Architecture Search

Neural networks have started to achieve state of the art results in several areas, such as Computer Vision, Natural Language Processing, drug design or time series forecasting, mostly after the so called deep learning revolution. Although the ideas of gradient descent, backpropagation, Convolutional and Recurrent Neural Networks had been around for years with promising applications, only after the year 2000 the use of GPU and new techniques for stable training of deeper networks unlocked the full deep learning potential. Key events were AlexNet winning the ImageNet competition in 2012 and the subsequent invention of ResNet.

The success of deep learning is due to the automatisation of tasks like data preparation or feature extraction that required human intervention for previous machine learning methods, because the network is able to learn directly from raw data. Better deep learning performances have been driven mainly by new components, like the transformer, and new architectures, more and more deep and complex. With time the number of different parameters that defines a network exploded, while designing new architectures remained a trial-and-error based endeavour. In fact, although general guidelines on what's best for every task exist, there is no actual scientific rule in creating new models.

Neural Architecture Search strive to make neural network design automatic. This is accomplished creating optimization algorithms, called *search strategies*, to explore the *search space*, the set of all the possible architectures.

So the search space is the set of all the valid values of all meta-parameters defining the architecture to be optimized.

One of the first search strategy was to use another *controller* network, trained to output architectures and learn to generate the best combination of layers based on the performances of the child networks. To evaluate the child models' performances all of them need to be trained, for each gradient step of the controller, requiring a prohibitive amount of GPU hours. Followup works focused on more efficient search algorithms, applying algorithms of reinforcement learning like *NAS-RL* [52].

Another line of work are the training-free methods, also known as zero-shot learning, that try to estimate generated architectures through lightweight score functions, without training at all [53].

Score functions could be designed based on theoretical or empirical findings. Some of them use the data that will then be used for actual training while others don't, so approaches can be divided in data-dependent or data-independent. They can also be divided in dependent or not from a specific starting model [54]. The effectiveness of training free metrics is usually evaluated measuring the correlation between the predicted performance of the generated model and its actual accuracy on test data after complete training.

A technique alternative to zero-shot NAS involves predicting final accuracy after a few training epochs, in order to save on training time.

Inspired by the previous ideas, a growing field seeks to develop automatic NAS algorithm where a controller discovers new architectures by searching for the optimal sub-graph within an over-parametrised network. The controller is trained to select child models that minimize error on the validation set. This is known as the one-shot technique, as the large model is trained only once, while its weights are shared between the child models.

This has been done in the Efficient Neural Architecture Search paper [55], but the complete pipeline still remains resource intensive and time consuming, regardless of whether a controller network or a reinforcement learning policy is used.

3.5.1 Differentiable Neural Architecture Search

Relaxing the previous constraints Liu et al. in *DARTS* [56] made the search space continuous, so it can be optimized by gradient descent. In this way the network weights and the meta parameters defining the architecture are jointly learned throughout a single training of the model. This method is called Differentiable Neural Architecture Search and the concept at its base could be used to rapidly generate networks meeting any possible needs. The most sought after targets are better accuracy or more lightweight models, the latter usually to comply to the constraints posed by edge devices, especially in terms of occupied memory and number of operations for real-time applications.

3.6 PLiNIO

PLiNIO [1] stands for Plug-and-play Lightweight tool for Deep Neural networks Inference Optimization. It's a package for gradient-based optimization that includes 3 different algorithms:

- SuperNet: a DNAS algorithms inspired by DARTS, often used for the coarser architecture exploration.
- Pruning In Time (PIT) [57]: a strategy for convolutional layers geometry optimization. It can do channel pruning, filter size pruning and dilation increase.
- MPS [58]: a differentiable mixed precision search tool for channel-wise precision optimization and pruning. In this work it has been used with a single precision choice, 8-bit integers, to perform Quantization-Aware Training (QAT).

3.6.1 SuperNet

The SuperNet subpackage allows us to define every layer of the network as an ensemble of different alternatives to be explored. A scheme of SuperNet setup and use is shown in figure 3.1.

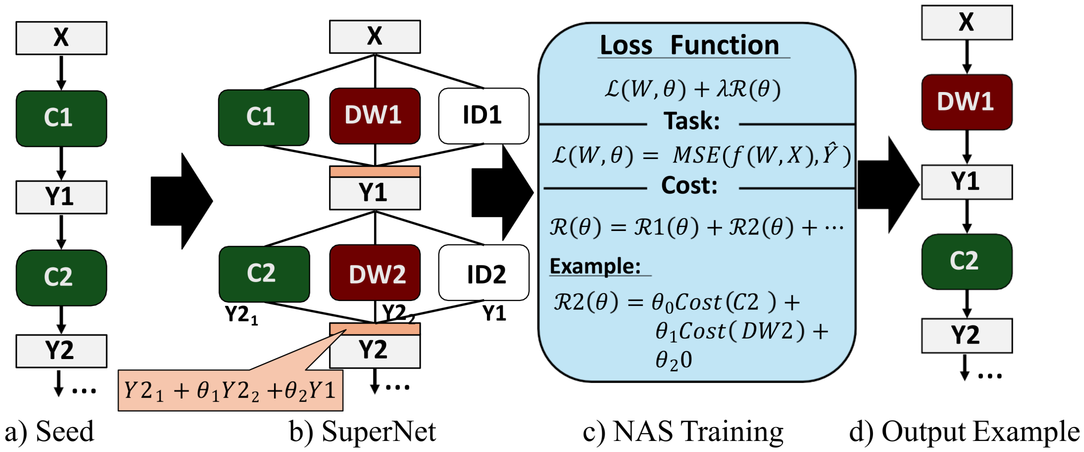
All alternatives must receive the same input, the output of the layer is the linear combination of all alternative's output, each of them weighted by an *architectural parameter* (θ_i).

These parameters are a continuous relaxation of the problem of choosing a single alternative, that correspond to setting one of them to 1 and all the others to 0. Throughout the training SuperNet optimise all of the θ jointly with the weights through gradient descent, then samples a different architectures at each epoch, applying Softmax. Thus it learns which alternative maximise the tradeoff between accuracy and cost for each layer.

The cost can be defined as the number of parameters, the number of operations or any other custom metric, and is added to the training loss, weighted by a *strength* parameter λ. This modifies the training loss during NAS as follows:

$$min_{W,\theta}\mathcal{L}(W;\theta) + \lambda\mathcal{R}(\theta) \tag{3.1}$$

In this work SuperNet has been used to explore the use of depthwise-separable convolutions and the overall network depth, using the identity function among the possible branches.


X
C1
Y1
C2
Y2
...
X
C1
DW1
ID1
Y1
C2
DW2
ID2
$Y2_1$
$Y2_2$
Y1
Y2
$Y2_1 + \theta_1 Y2_2 + \theta_2 Y1$
...
Loss Function
$\mathcal{L}(W,\theta) + \lambda\mathcal{R}(\theta)$
Task:
$\mathcal{L}(W,\theta) = MSE(f(W,X),\hat{Y})$
Cost:
$\mathcal{R}(\theta) = \mathcal{R}1(\theta) + \mathcal{R}2(\theta) + \cdots$
Example:
$\mathcal{R}2(\theta) = \theta_0 Cost(C2) + \theta_1 Cost(DW2) + \theta_2 0$
X
DW1
Y1
C2
Y2
...

a) Seed b) SuperNet c) NAS Training d) Output Example

Figure 3.1: SuperNet-based NAS, from [59]

3.6.2 Pruning in Time

The Pruning In Time [60] (PIT) tool allows the optimization of all the most relevant parameters of a convolutional layer. While SuperNet belongs to the class of the *Path-based* DNAS methods, PIT is a *Mask-based* DNAS, because it applies masks to weights and activation tensors of the model. These mask are optimized with gradient descent together with the weights, then binarized. At the end of the search the masked parts are eliminated, producing a shrinked version of the original architecture, called *seed*. This approach reduces the NAS cost as it searches only among DNN that can be obtained reducing the original one. It effectively performs structural pruning, the action of removing the less meaningful channels while maintaining the maximum possible accuracy and reducing the given cost.

Compared to SuperNet, PIT explores a narrower search space but carries on a much finer optimization. On 1-dimensional convolutions, our case, it can optimize the receptive field and dilation too. PLiNIO extends the original PIT adding support for 2D convolutions, the implementation is visualized in figure 3.2.

3.6.3 Mixed Precision Search

The Mixed Precision Search subpackage of PLiNIO use gradient descent to assign an integer precision for the quantization of each part of a model. It can be used to assign different precision to weights and activations, choosing from a list of configurable options. Quantization is a technique designed to reduce the size and computational costs of neural networks by storing the weights and

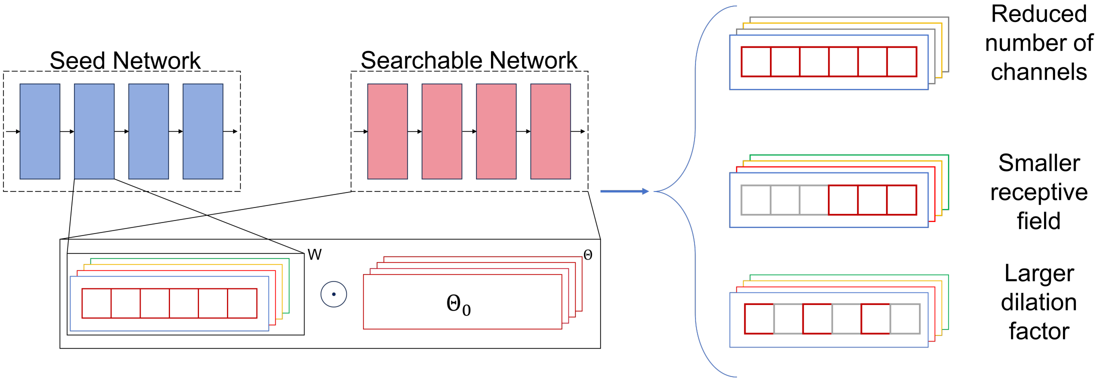


Figure 3.2: PLiNIO implementation of PIT on 1D convolution, from [60]

activations with lower precision data types, for example 8-bit integers instead of the commonly used 32-bit floating point numbers. The easiest conversion method is post-training quantization, where the network's weights are adjusted to the target format using a rounding function. This method quickly reduces the model's size, but it introduces a quantization error that lead to a significant drop in accuracy.

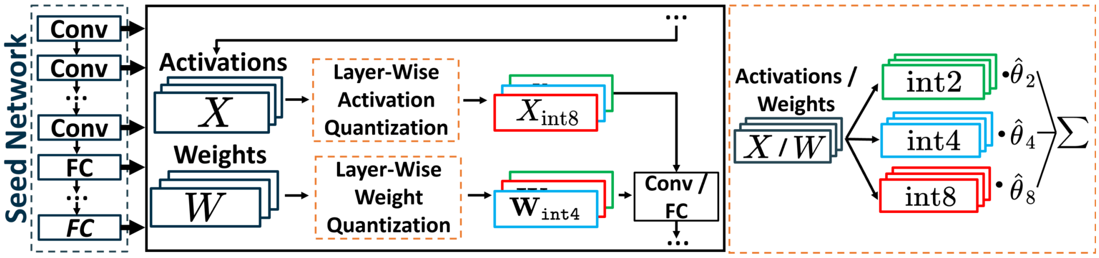


Figure 3.3: MPS-based Quantization, from [1]

In order to minimize the loss in accuracy, Quantization Aware Training (QAT) emulates the effect of quantized model during some additional epochs of training, so that the network can learn to compensate the added noise.

MPS has been used in this study to quantize the best models on the UCI dataset. A single precision, 8-bit integer format was set as the only available choice for each network internal parameters, meaning no actual precision search was performed. Instead, the tool carried out QAT, generating reduced CNN while minimizing the drop in accuracy.

Chapter 4

Methodology

4.1 Neural network seed architectures

This section will describe the three architectures used as starting point for the whole NAS process, underlining their specific peculiarities and the differences between them. A more extensive explanation of the application of NAS starting from these base architectures is in the section 4.3.

4.1.1 ResNet

ResNet is a deep CNN architecture, first introduced in 2015 for the ImageNet challenge [61]. The name comes from residual connections, i.e. the practice of adding the input tensor to the output of a block of layers. The blocks are usually repeated multiple times following a general scheme, in order to rationalize the network design. Residual (or skip) connections stabilize the training and unlock the potentialities of architectures way deeper than before.

In the benchmark, and consequently in this work, a 1-dimensional version of the original ResNet architecture is used, directly on single dimensional PPG signal.

The whole architecture is exposed layer by layer in figure 4.1. The architecture is composed of a first convolution that receives a univariate PPG input, followed by Batch Normalization, Rectified Linear Unit (ReLU) and a max pooling layer, then a block containing 2 different convolutional layers is repeated for a number of times that can be modified by a specific parameter. At the end the embeddings pass through a linear layer that performs the final regression predicting two output values, the systolic and diastolic blood pressure. Within the repeated blocks, right before the skip connection addition, an additional component known as squeeze-and-excitation is incorporated, first introduced in SENet

[62]. SENet ranked first place at ILSVRC 2017 Classification competition. The Squeeze-and-Excitation component explicitly models the interdependencies between channels and recalibrates their outputs. This mechanism begins with an average pooling layer, which compresses each channel into a single value – a process referred to as the *squeeze* operation. This is followed by two fully connected layers interleaved with ReLU and Sigmoid activation functions, forming the *excitation* operator. The first fully connected layer acts as a bottleneck, reducing the input, while the second one increase the tensor dimensionality back to the original number of channels. The *excitation* component use the input reduced by the *squeeze* to learn a set of channel weights that are then expanded and multiplied to the original input.

The values inside square brackets in figure 4.1 are architectural parameters that change for each dataset, the values inside correspond to PPGBP, UCI, BCG and SENSORS dataset, respectively. This happens because both this seed and U-Net have already been optimized [2], but only for accuracy. In contrast, in this work, some *cost-aware* optimizations are performed, showing that this can lead to similarly or better performing models, which are additionally smaller and more efficient.

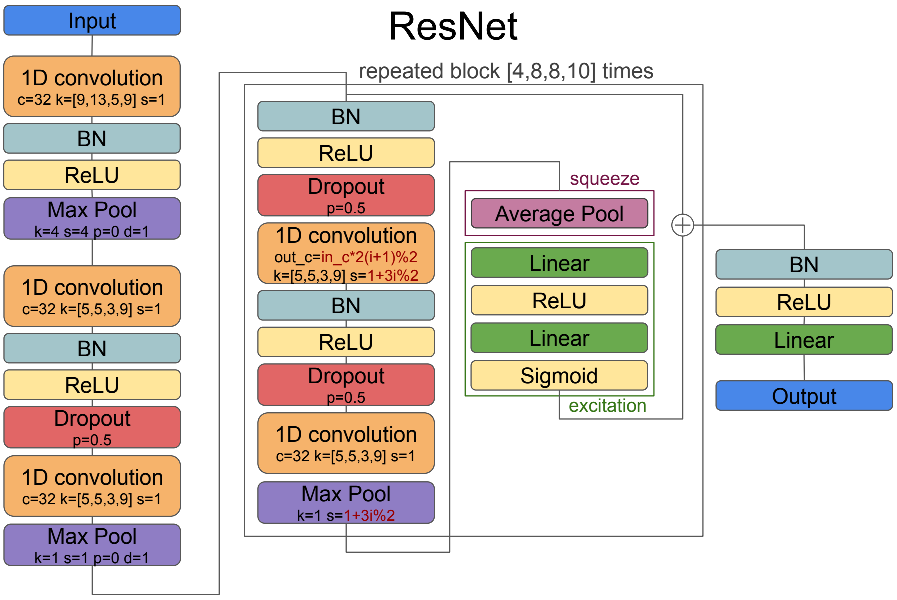


Figure 4.1: Scheme of the 1-dimensional ResNet architecture

4.1.2 UNet

U-Net is an encoder-decoder architecture developed for biomedical image segmentation at the University of Freiburg. The architecture is fully convolutional, thus it doesn't use any fully connected layers [63]. The architecture is composed of two main parts:

i) an initial contracting network which reduces the shape of the input sensors creating lower dimensional embeddings

ii) a symmetric decoder that restore dimensionality of the flowing data using upsampling layers.

In the decoder the upsampled data gets concatenated with the output of the corresponding layer in the encoder. The symmetry between the two descending and ascending branches makes the architecture visualization resemble an U letter, as visible in figure 4.2.

U-Net has been used also for denoising applications in many modern diffusion models and for prediction of protein structures. We used a variant of the architecture adapted to work on temporal data, using 1-dimensional convolutions. The U-Net generates an output tensor with a single channel and the same shape of the input PPG. This encoder-decoder architecture is trained to reconstruct the corresponding ABP sample, using mean squared errors between the continuous waveforms as a loss. During the evaluation and test phases the values of the systolic and diastolic blood pressure are extracted from the generated ABP signal detecting peaks and minima.

The diagram in figure 4.2 show the basic U-Net architecture, determined by a list parameter $[n_0 ... n_i ... n_N]$, where each element determines the number of convolutions inside each block and the list total length determines the number of blocks, so the depth of the network. Increasing the number of blocks the skip connections from the encoder to the decoder increase correspondingly. The setting *out_ch* in dark red in figure 4.2 defines the number of internal channels output of the first convolution. These parameters differs for each dataset, as they have been optimized to generate the best performing architecture. For more informations about the architectural hyperparameter optimization consult [2].

This optimization generated a deeper UNet architecture the larger the target dataset. The number of output channels of the convolution is doubled at each block of the encoder and halved in the decoder. The upsample layer is omitted in the last block of the decoder. The architecture use Instance Normalization layers, an operation that slightly differs from Batch Normalization. While the latter normalize data across the whole batch, the former normalize each signal in the batch independently. U-Net use as activation the parametric rectified linear unit (PReLU), a function similar to a leaky ReLU, where the slope of the function for negative values is a parameter trained independently for each channel.

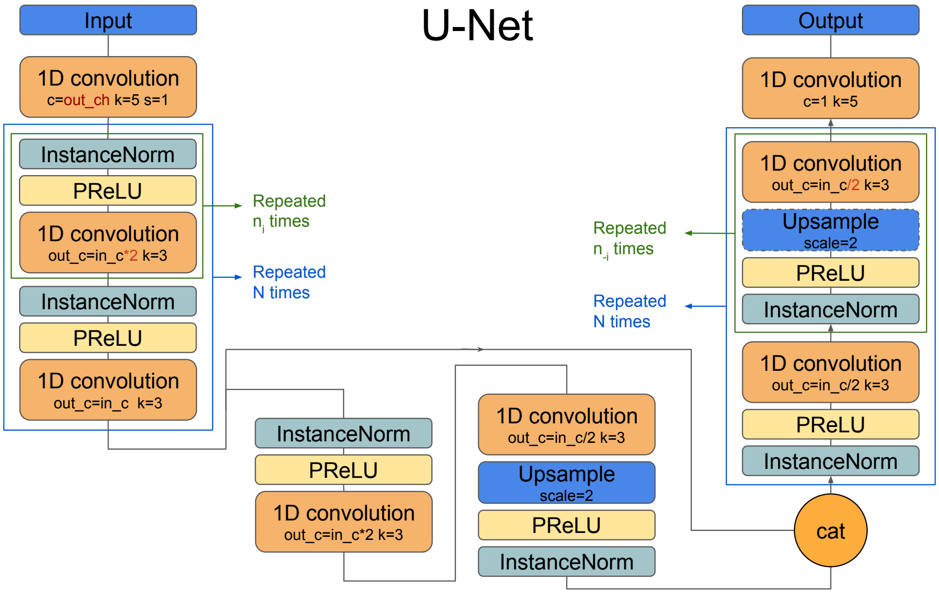


Figure 4.2: Scheme of the 1-dimensional U-Net architecture used

4.1.3 TEMPONet

Within the deep learning approaches 1-dimensional convolutional networks like ResNet show a good balance between accuracy and weight, so other similar architectures may perform equally good and be good candidates as seed, on which to apply the whole NAS pipeline. For this reason we added a Temporal Convolutional Network, TEMPONet, to the benchmark. TEMPONet has already been used for heart rate estimation from PPG, proving it can extract meaningful embeddings from this kind of temporal data. The TEMPONet architecture is presented in figure 4.3

The first test carried out, building up on the benchmark [2] codebase, aimed at reproducing the declared results for the ResNet and UNet models and evaluating TEMPONet trained in the same condition, as will be exposed in chapter 5.

TEMPONet consists of a series of repeated blocks. Each block's basic components include a 1-dimensional convolution followed by ReLU activation and Batch Normalization. Notably, it does not use residual connection within the blocks. In the last two blocks the convolution is replaced by fully connected layers that receive as input the output of the last convolution flattened.

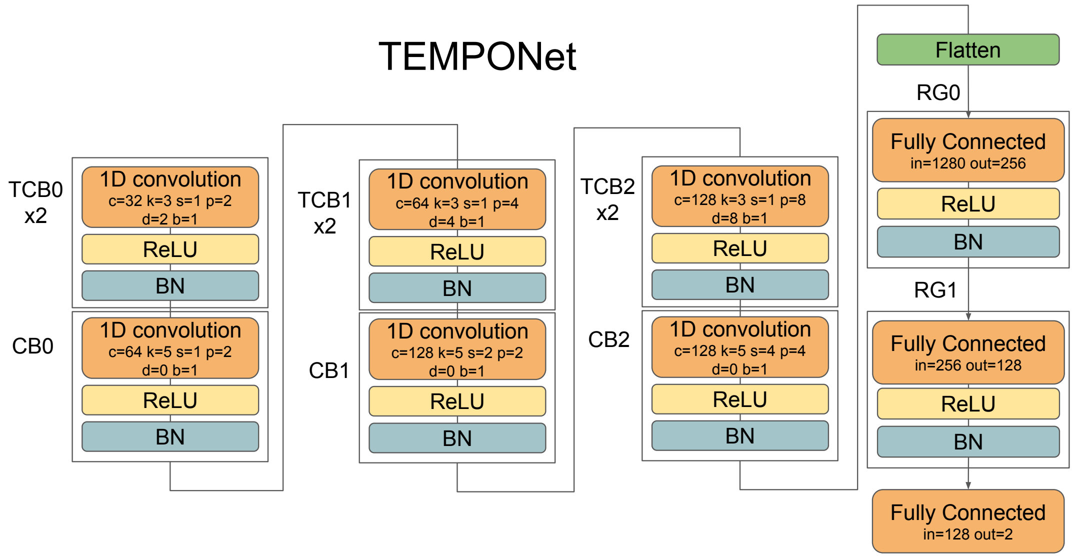


Figure 4.3: Scheme of the TEMPONet architecture

4.2 Data augmentation

Our 4 datasets are relatively small, considering the amount of data used to train neural networks for the tasks where deep learning excel. In order to lower our error metrics we experimented with data augmentation, evaluating the impact of data transformations that have already been useful for the heart rate task [11] on PPG signals.

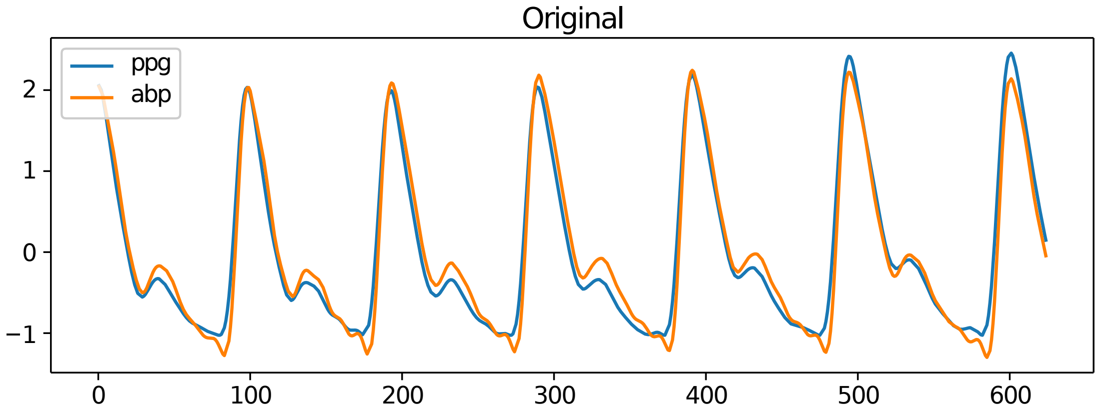


Figure 4.4: Visualization of the unmodified PPG and ABP signals superimposed

To do this some quite simple data transformations have been applied to all the datasets separately:

- **Jittering**: Adding at each timestep some gaussian noise, sampled from a gaussian distribution zero mean centered, with a tunable standard deviation. The resulting signal becomes as in figure 4.5. The relative mathematical expression is:

$$Xaug_{\mathrm{t}} = X_{\mathrm{t}} + \mathcal{N}(0, \sigma)\,\forall t \tag{4.1}$$

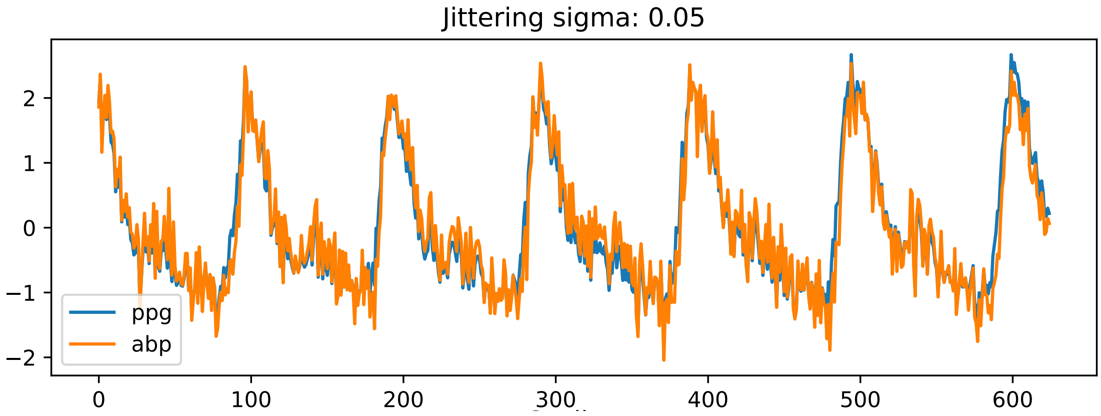


Figure 4.5: A visualization of the jittering transformation

- **Scaling**: This transformation consist in multiplying the whole sample for a random value, sampled from a gaussian distribution with $\mu = 1$. This method influence the final signal amplitude:

$$Xaug = X \times \mathcal{N}(1, \sigma) \tag{4.2}$$

  The effect of scaling a sample are reported in figure 4.6

- **Magnitude Warping**: This transformation also alter the magnitude of the signal. All samples are multiplied with a cubic spline interpolating randomly generated points, expressed mathematically:

$$Xaug = X \times CubicSpline(0...T, \mathcal{N}(1, \sigma)) \tag{4.3}$$

  The results are shown in figure 4.7

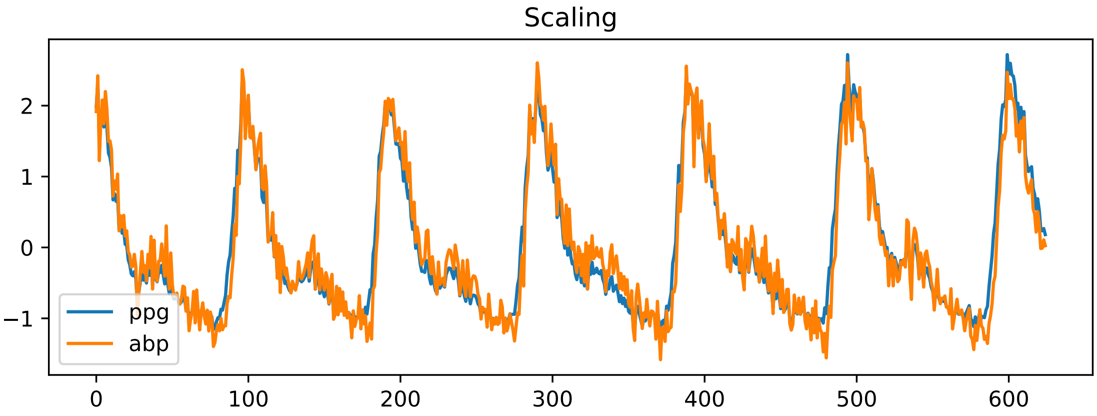


Figure 4.6: A visualization of the scaling transformation

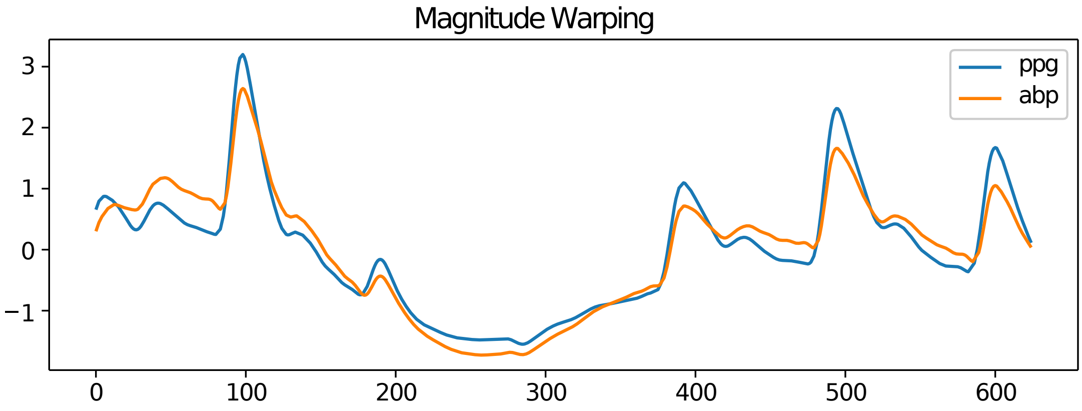


Figure 4.7: A visualization of the Magnitude Warping transformation

- **Time Warping**: The signal here is modified in time instead. This transformation acts on the distance between two data point, making them closer or further than the starting time step of $\frac{1}{f_s}$. The new distances are $CS_{\text{t}} - CS_{\text{t-1}}$, where CS_{t} are again points from a cubic spline:

$$Xaug = Interp(Cumulative(CubicSpline(0...T, \mathcal{N}(1,\sigma))), X) \tag{4.4}$$

The resulted signal becomes squashed in same places and extended in others, as shown in figure 4.8.

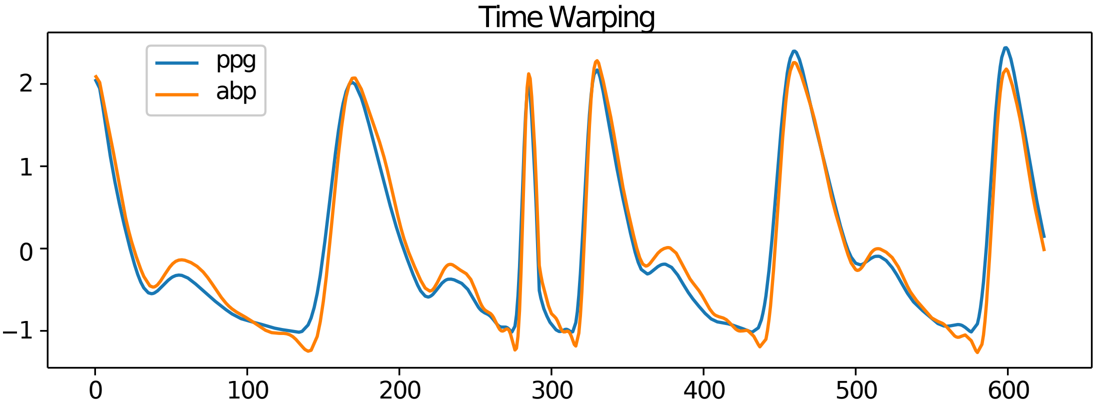


Figure 4.8: A visualization of the time warping transformation

4.3 PLiNIO

All three subpackages of PLiNIO, described in 3.6, were used for our Neural Architecture Search and optimization study. An important preliminary step in the search pipeline is selecting the starting architectures, called *seeds*. These will be presented in-depth in section 4.1. Then our optimization pipeline is applied, as illustrated in figure 4.9. The first gradient-based NAS implementation is SuperNet, it carries out a coarse-grained neural architecture search evaluating several variations of the original seed. The best models found with SuperNet are modified in order to apply Pruning-In-Time, to further optimize the models at a lower level and reduce their weight by pruning the less relevant data paths. Finally the best models undergo the Quantization Aware Training phase, executed through PLiNIO subpackage MPS. For some models we also estimate how they perform when deployed on hardware.

All PLiNIO experiments involved a pretraining phase, where the seed model is trained shortly without changing its architecture. The pretraining lasted always 20 epochs, unless the algorithm is applied starting from an already tuned checkpoint from a previous phase of the pipeline, in which case it’s omitted.

After the pretraining the architecture is converted to comply to each specific algorithm. PLiNIO automatically traverse the network as a graph mapping input and outputs of each layer, then solves the non-trivial problem of adding the correct NAS parameters θ on the correct edges.

The added parameters for gradient-based NAS or pruning are optimized on the validation set, keeping the model weights frozen. At the end of the NAS epoch a new model is sampled, the θ are frozen, and the network’s weights are trained normally on the training set.

In this way, we train a complete model only on the training data, while using the validation data for searching the best architecture, serving as a proxy of the test set and the real world. The two weights use different optimizers with different learning rates, they both use mean squared error as training loss.

During the training of the NAS parameters the loss is modified adding another cost to be minimized. To lower the model memory footprint our cost was the total number of parameters, multiplied for a value λ called strength. The strength has the dual purpose of adjusting the cost to the order of magnitude of the loss and defining how much the running algorithm should pursue a lighter model against a more complex and accurate one. The training phase has been set always to a minimum and a maximum of 50 and 160 epochs respectively. Additionaly, to prevent the training from being too short, an early stopping callback with a patience of 40 epochs was employed.

At the end of the training the model is automatically exported generating the reduced architecture. This final sampling happens through binarization of the parameters for the NAS. The alternative with the highest parameter is chosen in SuperNet, while only the channels of the kernel with a θ above a certain threshold are kept in PIT.

At the end of the search the new model retains the surviving weights, that have been already trained. To regain its accuracy after the binarization it is further fine-tuned for other 200 epochs. Only at the end the model is evaluated and compared against all the previous ones.

For all the experiments we kept the training parameters fixed, in order to compare all the models in the same conditions. A further hyperparameter optimization ablation study for the best models generated could improve the error metrics even more. All models used two Adam optimizers, one for the network weight, with learning rate set to 0.001, and another for the parameters of the NAS, with learning rate set to 0.01.

To better explore the accuracy-weight tradeoff landscape, several (9 or 18) values of strength have been employed, equally spaced in a logarithmic scale, from 10^{-11} to 10^{-7}, unless otherwise specified.

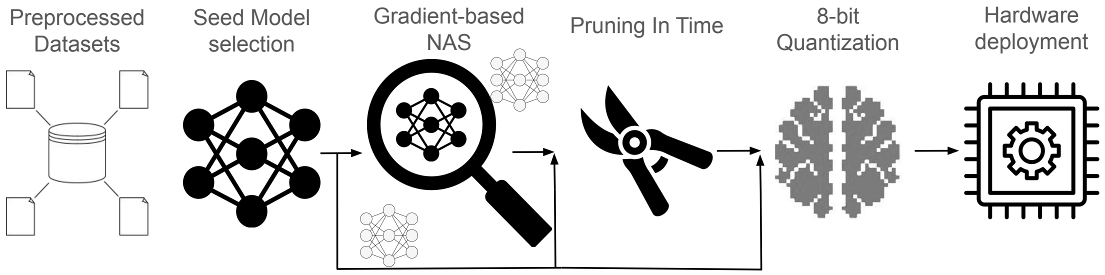


Figure 4.9: Visualization of the applied optimization pipeline based on PLiNIO

4.3.1 SuperNet

SuperNet, as a gradient based NAS, needs a limited number of defined alternatives of each layer to explore all possible combinations of choices. During training all the alternatives in each pool receives the same input, then the output is obtained as a linear combination of all the outputs of the single alternatives, each weighted by a softmax-ed trainable parameter θ_i.

The final architecture is obtained setting one of the $\theta_i = 1$ (and the others $= 0$). In order to use gradient descent SuperNet solves a relaxed continuous version of this problem: it inserts the DNN including the alternatives in a standard training, learning both model weights and new θ parameters jointly. In this work SuperNet has been applied to all 3 selected seeds (U-Net,TEMPONet and ResNet) using as alternatives for every convolutional layer:

- The original 1D convolution
- An equivalent depthwise-separable convolution
- An identity operation

The depthwise block is made of a depthwise convolutional layer followed by a pointwise convolution. It was first popularized by the MobileNet architecture [64], a lightweight model designed for constrained mobile devices. The depthwise convolution is a lighter yet similarly accurate approximation of standard convolutions, so it is a useful tool for the design of tiny but still capable networks.

The identity operation is added to the possible SuperNet choices only when the block input and output tensors have the same shape, allowing the NAS algorithm to remove completely some layers, exploring the impact of network depth.

4.3.2 Pruning In Time

In the second optimization step of our pipeline, some of the Pareto-optimal DNNs generated by SuperNet were selected for each dataset, to perform the further fine-grained search, using Pruning-In-Time. Using PIT we optimized number of features, kernel size and dilation of every convolution, all at once in a single training. PIT can freely explore all possible values for these parameters, without needing a set of values of choice as definite search space: at the end of the training phase the mask gets binarized and only the values above the threshold survive.

4.3.3 Quantization with MPS

Some of the best models generated by SuperNet underwent Quantization Aware Training to `int8` format, to evaluate the network inference capability on edge devices having integer arithmetic only. Quantization was implemented through PLiNIO subpackage MPS, using a standard min-max affine quantization format for all the weights and the Parametrized Clipping Activation (PaCT) method for layer's inputs and outputs [65].

MPS generated quantized models entirely on 8-bit integers, ready for deployment. Accumulation and biases are on 32 bits, as supported by our target inference library PULP-NN[66].

Chapter 5

Experimental Results

5.1 State of the art model footprint evaluation

The first step in our twofold exploration is to evaluate all the previous state of the art model, considering the size of the model besides accuracy. To do that, we added the information on number of parameters and number of operations to all the deep learning models already present in the aforementioned benchmark, and to two machine learning approaches: support vector machines and random forest.

The number of parameters and operations of the SoA models are reported for each dataset in table 5.1.

The tradeoff between accuracy and resource consumption is summarized in figure 5.1. On the x axis there are the number of parameters in logarithmic scale, on the y axis the mean absolute error for the systolic (above) and diastolic (below) blood pressure. The size of the marker is proportional to the number of operations.

We can see how the two traditional machine learning approaches considered are the absolute best in the smallest dataset, PPGBP, and are on the pareto front of all the other datasets too. Among the deep learning models ResNet and UNet performs consistently well on all 4 datasets, for this reason they have been the main seed of our NAS investigation. The model MLPBP has average performances but a very large number of parameters, for this reason it will be left out of the next plots.

Table 5.1: Number of parameters and operations of the state of the art models

Model	PPGBP			BCG		
	# parameters	SBP MAE	DBP MAE	# parameters	SBP MAE	DBP MAE
SVR	SBP: 29,05k	13.15	-	10,70k	11.45	-
	DBP: 16.87k	-	8.04	55,89k	-	7.34
RF	SBP: 20,27k	13.17	-	210	12.88	-
	DBP: 20,36k	-	8.12	85,25k	-	7.89
ResNet	49,35k	13.402	8.451	486,34k	11.945	7.895
SpectroResNet	241,51k	18.87	11.38	251,30k	12.41	8.3
MLPBP	6,70M	16.49	8.8	28,44M	12.39	8.05
UNet	-	-	-	446,72k	12.3	7.98
PPGIABP	-	-	-	296,40k	17.06	8.07
VNet	-	-	-	491,55k	11.42	8.01
Model	SENSORS			UCI		
	# parameters	SBP MAE	DBP MAE	# parameters	SBP MAE	DBP MAE
SVR	SBP: 775,34k	15.60	-	18,46M	17.45	-
	DBP: 415,77k	-	7.50	4,54M	-	8.07
RF	SBP: 64,04k	15.86	-	21,34k	16.85	-
	DBP: 170,79k	-	7.66	4,26k	-	8.25
ResNet	1,93M	17.46	8.33	791,75k	16.588	8.298
SpectroResNet	251,30k	17.83	8.31	254,76k	19.88	9.0
MLPBP	28,44M	17.61	8.26	28,44M	17.57	8.38
UNet	140,87k	15.64	7.66	29,75k	16.93	7.88
PPGIABP	296,4k	16.45	7.99	296,4k	17.06	8.07
VNet	530,27k	16.77	8.62	2,17M	17.58	8.95

5.2 TEMPONet results

The first step in our deep learning exploration was the evaluation of TEMPONet architecture, described in section 4.1.3, training and testing it on all the considered datasets. The complete comparison with previous State of the art models is shown in figure 5.2, where TEMPONet is circled in red. TEMPONet achieved good results on PPGBP, matches the average performance of other models when trained on SENSORS and UCI but struggles considerably on the DBP task of BCG. The relatively shallow architecture of TEMPONet, compared to the other models, likely explains its strong performance on smaller datasets, while the other networks achieve a better understanding of the signals on larger datasets.

It's worth noting that in all datasets it failed to outperform the best models, in particular it proved to be always inferior to ResNet. As shown in section 4.1.1, ResNet popularized residual connection as a means to stabilizes training and allow convergence of deeper models. In fact depth has been the key to its success in many tasks since its initial conception. ResNet is, indeed, deeper than TEMPONet, including more parameters but requiring less operations, thanks to

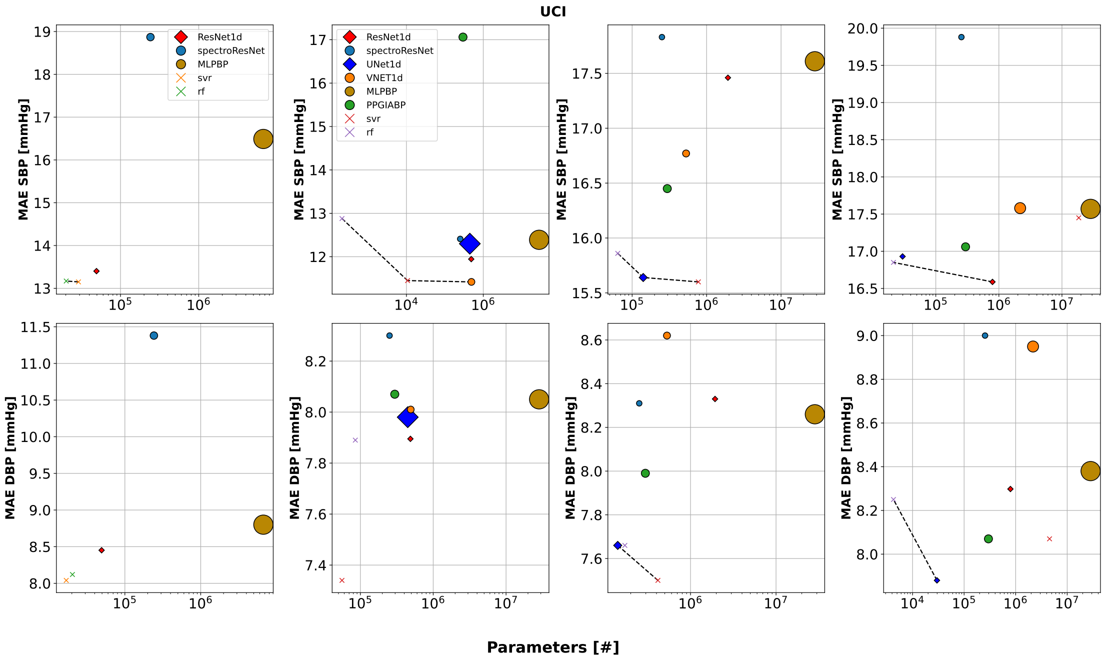


Figure 5.1: Summary of the state of the art models from the benchmark.

an initial stem network that drastically reduce input dimensionality in the first layer.

5.2.1 TEMPONet variants and the effect of dropout

In an attempt to better explore TCN potentialities, three reduced variations of TEMPONet were created, hoping to obtain smaller models with accuracies close to those of TEMPONet. They are called R1, R2 and R3, their size in number of parameters and operations can be seen in table 5.2.

These variants were created by adding configuration settings to the script that defines the TEMPONet architecture. These settings include the number of basic blocks to adjust the network depth, as well as lists of values to specify the receptive fields, the number of channels, and the dilation of each convolutional layer. Furthermore, a stem layer was introduced at the beginning of the R3 architecture to reduce the dimensionality of the input tensor. That makes this model the least computationally intensive, though it comes with a slight increase in the number of parameters compared to R1. All these reduced alternatives show worse accuracies than the original TEMPONet, especially for SBP.

A close examination of the learning curves revealed that all TEMPONet architectures are more prone to overfitting, compared to ResNet. To enhance model

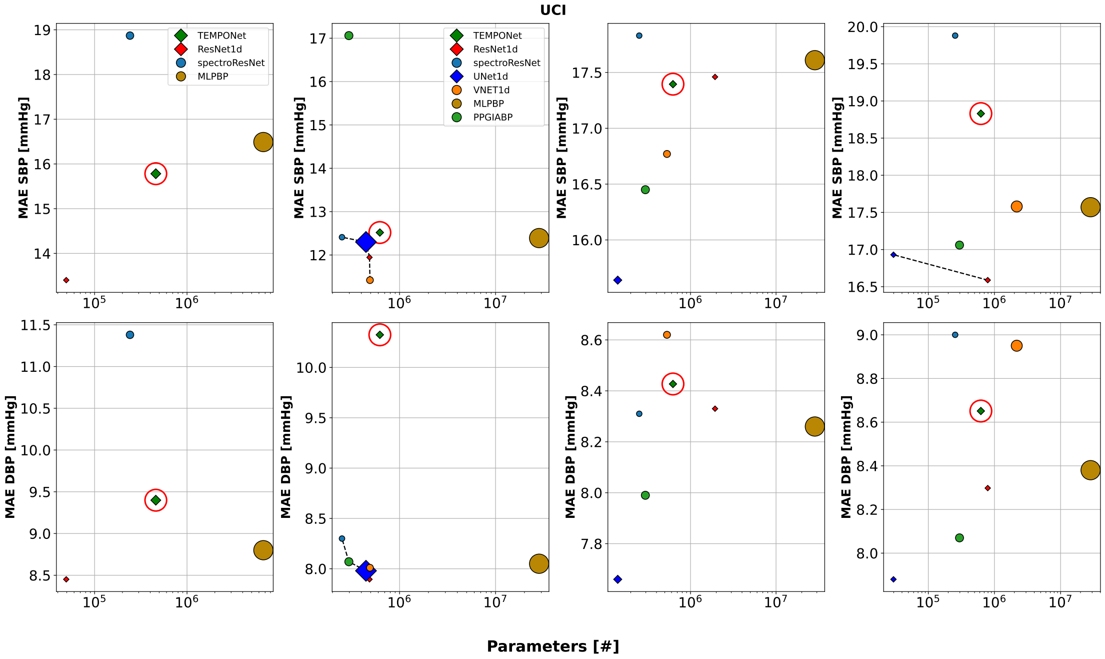


Figure 5.2: Comparison of all Deep Learning state of the art with TEMPONet, circled in red

generalization on the test set, we explored the use of regularization. Table 5.2 reports the results obtained adding dropout with value 0,2 before the last linear layers of each model. This approach samples a different architecture at each epoch, forcing the model to learn more robust features before the final blocks. It was not placed at the very end to allow the fully connected layers to correct errors introduced by random dropping before the final classification. Additionally, it was not applied to the hidden convolutional layers, as they are already followed by batch normalization.

The dropout layer had a slightly harmful effect on the original TEMPONet, but improved the accuracies of all the variants, bringing them very close to TEMPONet, and in one instance, even surpassing it.

This effect can be explained considering that smaller models like the variants should naturally overfit less, while TEMPONet probably require more sophisticated regularization methods.

Model	Parameters	Operations [Mmac]	without dropout		Dropout 0,2	
			MAE-SBP	MAE-DBP	MAE-SBP	MAE-DBP
TEMPONet	459490	13,77	15,782	9,399	16,234	9,856
R1	164226	8,05	17,004	10,221	16,353	9,413
R2	193282	6,43	17,383	10,101	15,697	9,426
R3	189194	3,2	17,933	10,645	17,144	9,539

Table 5.2: Sizes and errors of TEMPONet and its variants on PPGBP, with and without dropout

5.3 Impact of data augmentation

All the data transformations techniques outlined in section 4.2 were employed to generate two augmented dataset for each of the original four, with different percentages of augmentation. The transformations applied and the corresponding amount of synthetic data generated are summarised in table 5.3.

Table 5.3: Augmentation configurations explored for PPGBP, BCG and Sensors

Augmentation	Augmentation type	Parameter	Percentage
6×	Jittering	σ : 0,05	100%
	Jittering	σ : 0,2	50%
	Scaling	σ : 0,05	100%
	Scaling	σ : 0,2	50%
	Time Warping	σ : 0,05 knot: 4	100%
	Magnitude Warping	σ : 0,05 knot: 4	100%
9×	Jittering	σ : 0,05	100%
	Jittering	σ : 0,2	100%
	Scaling	σ : 0,05	100%
	Scaling	σ : 0,2	100%
	Time Warping	σ : 0,05 knot: 4	100%
	Time Warping	σ : 0,2 knot: 4	100%
	Magnitude Warping	σ : 0,05 knot: 4	100%
	Magnitude Warping	σ : 0,8 knot: 4	100%

The TEMPONet and ResNet architecture have been trained anew on the data obtained augmenting PPGBP, BCG and SENSORS datasets. The models have been validated and tested on the same sets as before, to correctly compare MAEs.

All results obtained are shown in table 5.4. They clearly shows how all our transformations marginally improved model's accuracies, generating the best values starting from ResNet. Considering the time and computational resources needed to train on the substantial amounts of data generated by DA, along with the unpromising results observed so far, the data augmentation experiment was not pursued further for the largest dataset, UCI.

Table 5.4: Data augmentation results

Dataset	Augmentation	TEMPONet		ResNet	
		SBP-MAE	DBP-MAE	SBP-MAE	DBP-MAE
PPGBP	Original	15,782	9,399	13,402	8,451
	6x	15,615	9,254	15,733	9,168
	9x	15,985	9,354	13,283	8,46
BCG	Original	12,518	10,324	11,945	7,895
	6x	13,174	8,753	11.971	8.423
	9x	13,67	8,921	12,094	7,682
Sensors	Original	17,395	8,427	17,46	8,33
	6x	17,75	8,654	15,865	7,625
	9x	16,306	7,844	15,846	7,602

5.4 SuperNet

After these preliminary experiments, we moved on to apply the proposed automatic pipeline previously described in section 4.3. The optimization starts with a preliminary coarse Neural Architecture Search, implemented through PLiNIO's SuperNet algorithm, to be later refined by Pruning-In-Time. SuperNet's starting points are the best state of the art models among the deep learning architectures in the considered benchmark, so the ResNet and U-Net based 1-dimensional CNNs. In the PPGBP dataset the UNet model is not applicable because the continuous Arterial Blood Pressure it should reconstruct is not present, so we applied SuperNet on the newly added TEMPONet model instead.

All the generated models are reported in a mean absolute error (MAE) vs model size (expressed in number of parameters) plane as shown in figure 5.3, together with the seeds and the state of the art deep learning models. On the x axis the number of parameters are reported on a logarithmic scale.

From the plot the worst model (MLPBP) and the traditional machine learning approaches have been omitted for better visualization. In the graph the seed models are represented as squares, the SuperNet generated models as diamonds with colors corresponding to the seed, all the other Deep Learning SoA models as circles. The Pareto-optimal architectures are connected by a black dashed line.

SuperNet generated, on all datasets, models that either dominate the seeds or are on the memory-against-error Pareto front. The NAS has generated a new best performing model, reducing the lowest reached MAE, on all datasets except the smallest one, PPGBP.

On PPGBP, we obtain a rich Pareto curve of architectures starting from ResNet. We are able to reduce the seed size by 16%, with a small increase in MAE of only 3.9% and 3.5% for DBP and SBP prediction, respectively.

On BCG, we Pareto-dominate both seed networks, improving both their MAE and size. Our best UNet-derived model obtains 11.139 mmHg MAE on SBP prediction and 7.52 mmHg MAE on DBP, being 6.7%/4.7% better than the best seed (ResNet). Simultaneously, this network reduces the total number of parameters by 3.8 times.

However, it is important to note that for these two datasets classical ML methods, such as Support Vector Regressor (SVR) and Random Forest (RF), still outperform SuperNet's DNNs in both performance and size, as reported by [2].

In fact the SVR achieves the lowest MAE in DBP estimation for both PPGBP and BCG datasets (8.04 and 7.34 mmHg, respectively) and a MAE of 13.15 mmHg and 11.45 mmHg on SBP estimation, being outperformed by UNet only on BCG. That could be explained by both datasets' limited size.

However, while being interesting for small datasets, classic ML models fail to benefit from the availability of larger amounts of data.

On the second largest dataset, Sensors, classical ML methods have slightly better performance than the seeds, but they are surpassed by the new NAS-optimized DNNs. Namely, SVR, which achieved the best results on both metrics (15.60 mmHg for SBP and 7.50 mmHg for DBP), is now outranked by our UNet NAS models (15.51 mmHg for SBP and the same DBP) with up to 40 times less parameters.

On UCI, the dataset with the most samples, classic methods are outperformed by the deep learning seeds, which was already shown in [2]. The best machine learning model, RF, achieves a SBP MAE of 16.85 mmHg (versus 16.59 mmHg of the ResNet), while SVR is outperformed by UNet, with a DBP MAE of 8.07 vs 7.88 mmHg respectively.

Moreover, the higher complexity of these datasets causes the number of parameters of both the SVR (that is using a RBF kernel) and of the RF to increase exponentially. For instance, on UCI, the SVR becomes 998 times larger than our best SuperNet output.

Conversely, on these two larger datasets, thanks to our NAS, we are again able to obtain Pareto-dominant solutions. On Sensors, our UNet-derived architectures reduce the size of the most accurate seed (UNet) by 3.4×, while achieving a similar or lower MAE of 7.51 mmHg / 15.51 mmHg on DBP/SBP, respectively.

Interestingly, on BCG and Sensors, Unet-based architectures outperform ResNets. We attribute this behaviour to the ability of this network topology to learn faster from a lower amount of data, thanks to the richer training signal provided by the full time series reconstruction task. The situation reverses in UCI, where ResNet-derived DNNs achieve the best performance.

The most accurate networks found with SuperNet on UCI require only 149.8k and 156.3k parameters to achieve a close-to-optimal MAE of 16.655 mmHg on SBP estimation, and the lowest overall (7.86 mmHg) on DBP estimation. While the seed ResNet is able to achieve an even lower MAE on SBP, with its 792k parameters, it would be impossible to deploy on GAP8's internal memory of 512KB, even when quantized, as explained in the next section 5.5.

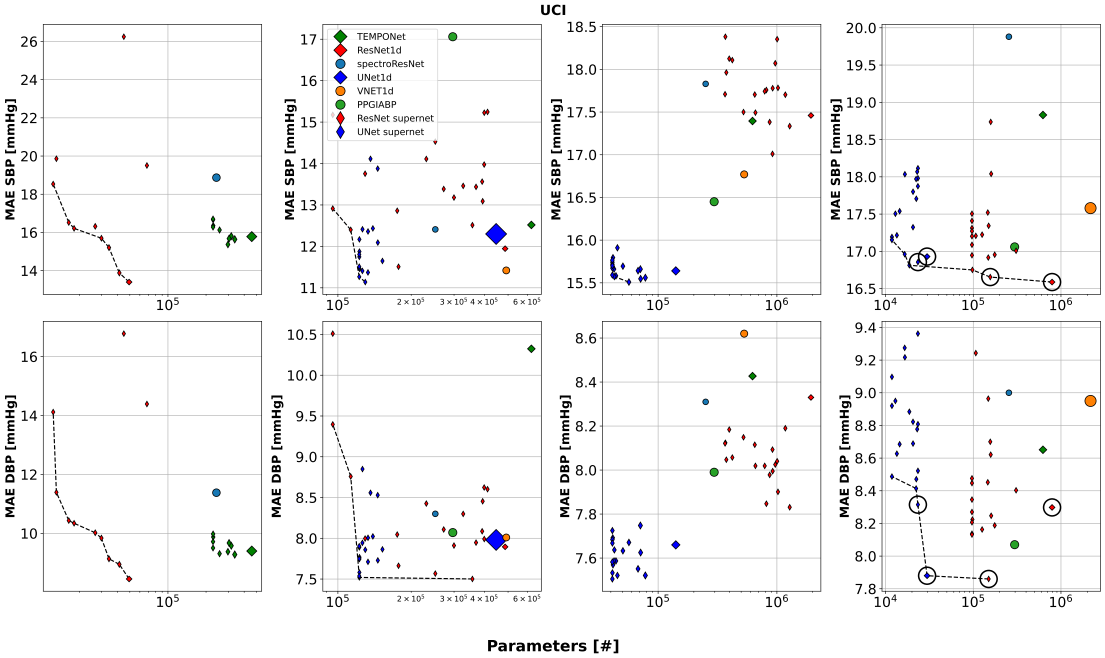


Figure 5.3: Results of the application of SuperNet on all datasets on DBP and SBP prediction.

5.5 Quantization aware training

The quantization step has been applied only on the biggest dataset, UCI. The four quantized models include the original ResNet and UNet architectures and 3 new Pareto-optimal topologies generated by SuperNet starting from them. The results of the quantization are summarized in table 5.5, where *ResNet-B* and *ResNet-S* are the biggest SuperNet models on the Pareto front of the SBP and DBP plots, respectively. *UNet-S* is instead a smaller architecture that is Pareto-optimal for both SBP and DBP tasks. The table reports the errors of each network on both tasks before and after quantization and their memory footprint, latency and energy consumption when deployed on the GAP8 platform.

It's worth noting that non-quantized models can't be deployed on GAP-8 because it doesn't have a Floating Point Unit. The glsMAE of fp32 seed models is thus reported solely to evaluate the accuracy drop due to quantization. Indeed, it can be observed a slight increase in MAE, up to 9.8%, with ResNet models being more affected by this degradation. The best results after quantization are achieved by ResNet-S on DBP estimation, with a 8.08 mmHg MAE, and by UNet-S on the SBP task reaching 17.2 mmHg MAE.

The seeds have higher errors and the original ResNet can't be deployed because its too high number of parameters doesn't fit GAP8's internal memory. On the contrary, all SuperNet models can operate within the platform's 512 kB memory threshold, with values of latency and energy consumption similar to the UNet seed. The highest values, a latency of 8.91 ms and an energy consumption as low as 0.45 mJ, are exhibited by UNet-S. This is probably because it is mainly composed of depthwise-pointwise convolution layers, which have less parameters but are usually less efficient after deployment, reducing memory consumption at the expense of a small rise in latency.

Table 5.5: Quantization results

Model	MAE-SBP	MAE-DBP	Size [B]	Lat. [ms]	E. [mJ]
Floating Point Models (fp32)					
ResNet	16.59	8.3	3.17M	n.a.	n.a.
UNet	16.93	7.88	118.9k	n.a.	n.a.
Quantized Models (int8)					
ResNet	18.23	8.17	791.8k	o.o.m.	o.o.m.
UNet	17.63	8.19	29.8k	7.04	0.36
ResNet-B	17.83	8.44	156.3k	7.12	0.36
Resnet-S	17.48	**8.08**	149.8k	7.27	0.37
UNet-S	**17.2**	8.26	23.4k	8.91	0.45

5.6 Pruning In Time

The second phase of the pipeline, Pruning-In-Time (PIT), continued the NAS exploration starting from the best DL models, including the ones generated by SuperNet.

On each dataset the best model overall and one or two others have been selected as seeds for PIT. The seeds that were generated by SuperNet are identified by a number postponed after the original architecture corresponding to the respective SuperNet experiment, sorted according to the strength parameter in ascending order. For example, the model UNet-7 has been generated starting from UNet with SuperNet, using the 7th strength parameter in the logarithmic scale.

The plot includes all PIT-generated models and the SoA CNN. All seeds are circled in black, the seeds obtained through SuperNet are depicted with a elongated diamond marker, colored matching the hexagonal PIT points obtained from them. UNet based PIT jobs use different shades of blue, while ResNet-based dots use shades of red. All marker's sizes are proportional to the number of operations required by the model at inference.

The plot shows how PIT further advanced the Pareto front on all datasets and created new best models on two of them, including the larger one, UCI.

On PPGBP the only seed used for PIT was the original ResNet architecture, because it was still the most accurate model. On this dataset PIT obtained a numerous group of new models on the pareto front with a better balance between memory and accuracy, even if none of them surpassed ResNet accuracy. Indeed, among the DL models ResNet has still the lowest MAE, but also a memory footprint larger than any NAS optimized model.

Noteworthy, the best models remains support vector regressor and random forest, the magnitude of the dataset is probably too small to fully exploit deep learning potentialities. The models generated by PIT here have 55% less parameters than ResNet with a 10,07% increase in SBP MAE, or a 52% parameter reduction with a 2.73% increase in DBP MAE. The results on the PPGBP dataset are reported in Figure 5.4.

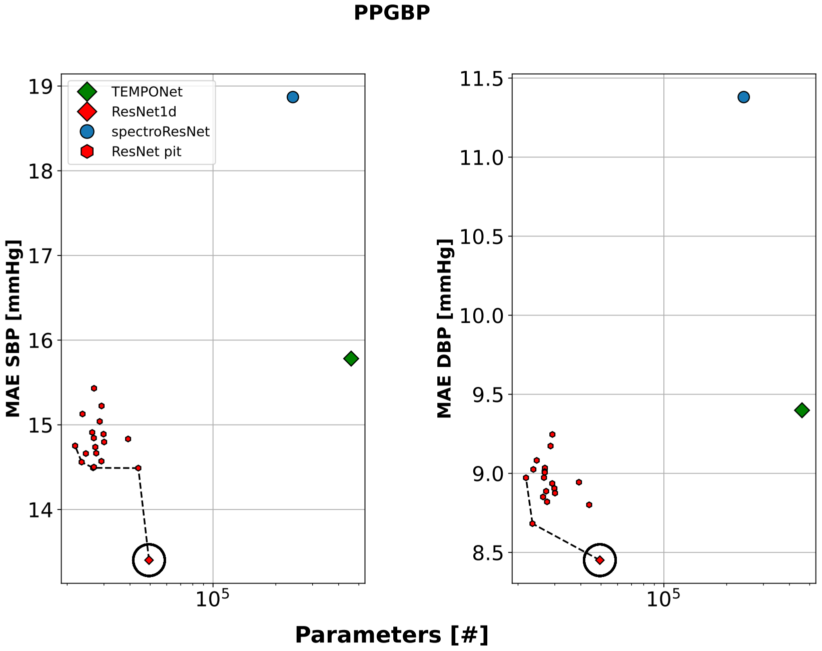


Figure 5.4: Results of the application of PIT on PPGBP on DBP and SBP prediction, seeds circled in black.

On BCG the new PIT models dominate all previous neural networks, redefining completely the Pareto front. The best models achieved a 4,676% reduction of error on SBP with 45,9% less parameters and a 7,99% decrease of DBP MAE with a 86,68% parameter reduction, compared to the best SBP and DBP models, VNet and ResNet respectively, as can be seen in Figure 5.5.

Other models achieved a 96,12% decrease in number of parameters while still having a 1,05% decrease in error on systolic pressure, or a 92,47% lighter model with a 1,77% diastolic pressure MAE decrease.

On this dataset PIT also managed to outperform classical ML methods that previously outperformed all other models, surpassing the limits showed by SuperNet. That could be explained by PIT's larger search space and finer optimization capabilities.

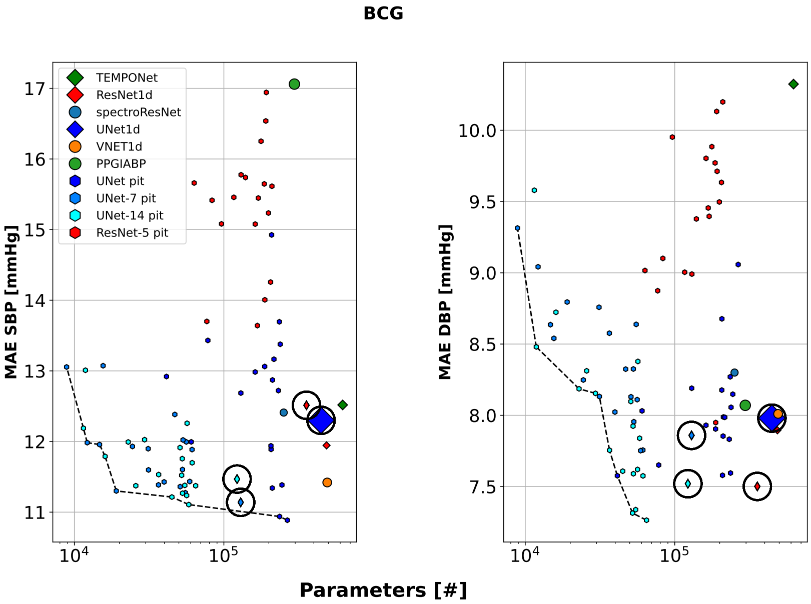


Figure 5.5: Results of the application of PIT BCG on DBP and SBP prediction, seeds circled in black.

On SENSORS, PIT created several models with drastically reduced memory footprint but also larger errors. Further experiments with better tuned strength ranges could create a continuous front, improving accuracy too. These lightweight models could be object of study for subject specific finetuning studies where they could achieve a very good compromise between low-consumption and finetuned accuracy. The best models generated by PIT achieved a 98,86% or 89,15% parameters reduction with an increase in SBP and DBP MAE respectively of 1,74% and 1,72%, compared to ResNet and SpectroResNet models. These models are not the best models because UNet and VNet outperforms them by far. Sensors' data seem particularly well suited to the Sig2Sig approach, as shown in Figure 5.6

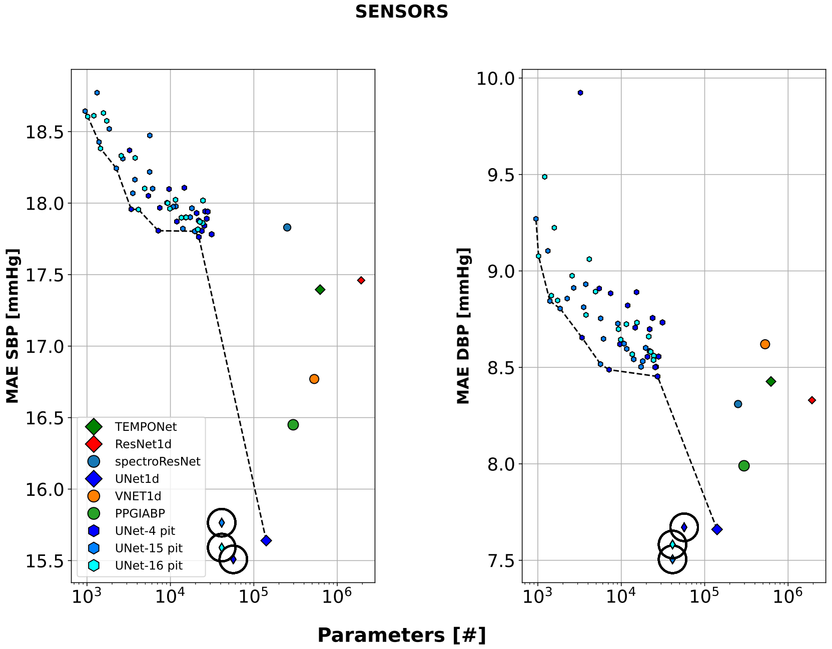


Figure 5.6: Results of the application of PIT on SENSORS on DBP and SBP prediction, seeds circled in black.

On UCI, PIT achieved the same remarkable feat realized on BCG: completely redefining the Pareto front, that is clearly visualized in figure 5.7.

As a matter of fact, new most accurate models were generated for both SBP and DBP. Moreover all models generated have less parameters than every SuperNet models. Here, PIT achieved a 1,59% accuracy improvement using 99,3% fewer weights on SBP prediction task compared to the previous best, ResNet. On DBP as well it reached a 2,31% lower error with a 71,67% more efficient model, compared to UNet, most accurate DBP neural network. The best models accomplish a MAE of 16,324 mmHg on SBP estimation and 7,698 mmHg on DBP. Compared to the best CNN among the generated models some candidates reach comparable accuracies (+0,79% and -1,65% MAE) being 99,67% and 91,7% lighter, for SBP and DBP respectively.

All dataset considered, our pipeline proved useful in obtaining better models overall, improving the State of the art, while also generating a wide set of models with different compromises between efficiency and accuracy. Even though the error values of all evaluated models are still far from clinical standards, this study proves that there room for improvement and provides various architectures that can deliver valuable BP estimations working in different low-power and memory constrained environments.

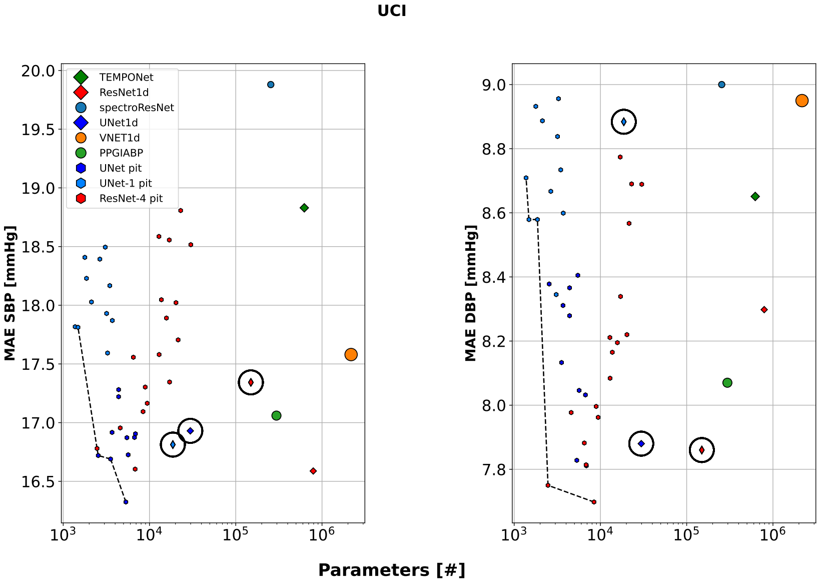


Figure 5.7: Results of the application of PIT on UCI on DBP and SBP prediction, seeds circled in black.

The results of the application of the whole pipeline to all datasets are summarized in figure 5.8.

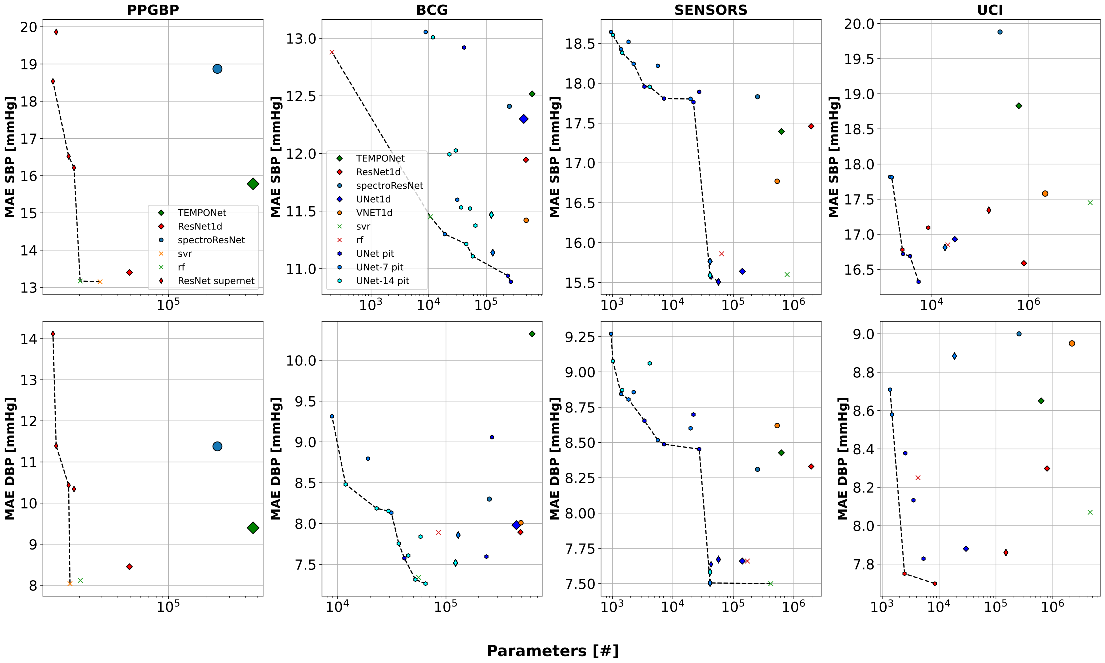


Figure 5.8: Results of the application of the proposed pipeline to all datasets, including Pareto dominant models, the respective seeds and state of the art

Chapter 6

Conclusions

This work showed that embedding accurate DNN models for BP estimation on low-power wearable-class devices is feasible, and the efficient and lightweight models found achieve state-of-the-art performances. Many models reached comparable or better accuracies than the original deep learning architectures using way less parameters. Some of the models generated with the proposed pipeline even improved the accuracies, achieving new all-time-low values of error. The workflow applied in this work proved capable of finding new optimized architecture, refine and quantize them, covering all the phases needed from the start to the final deployment. All the models generated through the pipeline fit GAP8 SoC platform memory after deployment, the corresponding values of latency and energy consumption shows significant savings.

The proposed pipeline proved its ability to develop a BP estimation system on a wearable. However, the primary challenge remains improving measurement accuracy, as medical applications are highly sensitive to errors. while the generated models can already provide the patient with useful insights regarding the trend of their blood pressure and thus serve as effective tools for preventing hypertensive conditions, they are not yet suitable as medical-grade instruments. In order to enhance performance and meet clinical standards, two main lines of research could be the object of future work:

- Apply more modern architectures, leveraging recent advancements in related fields, such as time series analysis and natural language processing
- Perform subject-specific fine-tuning of the most accurate general models to create personalized estimation methods with higher accuracy.

The first approach involves training and testing transformer-based ML algorithms which achieved success in many similar tasks. Transformers are particularly promising for BP estimation because:

- The PPG-to-ABP translation is a sequence-to-sequence problem, akin to multilingual translation, a category of tasks for which self-attention mechanisms were originally conceived.
- The complex, non-linear relationship between the PPG and ABP benefits for longer context windows. CNNs in this work elaborate a whole 5-seconds sample at a time, but using datasets with longer measurements could allow models to produce predictions increasingly aligned with the continuous ABP waveform.

Similarly, small language models trained to understand natural language sequences could potentially be adapted for these biosignals. Exploring the tradeoff between model size and error metrics on these models could yield significant benefits.

The subject-specific fine-tuning approach focuses on improving the precision of existing models instead, recalibrating them for single individuals. Many commercial products already require periodic recalibration using a precise sphygmomanometer. Practical implementation of this calibration necessitate using as few measurements as possible and avoiding reliance on the continuous ABP ground truth, as this data cannot be collected outside clinical settings. This reframes the problem as one of few-shot learning.

In the case of Deep Neural Network this calibration could be achieved through transfer learning, fine-tuning a general model trained on large datasets with a few PPG samples collected by users at home using a portable sphygmomanometer. Another approach could involve creating large and accurate models and then applying knowledge distillation towards smaller and more efficient networks that are easier to recalibrate for individual subjects.

Lastly, exploring the development of theory-guided ML models could be a fruitful direction. Existing mathematical models describing blood pulse as a fluid flowing in elastic pipes have limited application to PPG due to the signal's complexity.

Factors such as light interactions with tissues and the multiple reflected waves in the closed cavities of capillaries make a-priori models of PPG unattainable. Nonetheless, integrating physical and medical knowledge into ML models could restrict the solution space explored during training, potentially leading to more accurate estimations.

Bibliography

[1] D. Jahier Pagliari, M. Risso, B. A. Motetti, and A. Burrello. *PLiNIO: A User-Friendly Library of Gradient-based Methods for Complexity-aware DNN Optimization.* 2023. arXiv: 2307.09488 [cs.LG] (cit. on pp. 3, 26, 28).

[2] Sergio González, Wan Ting Hsieh, and Trista Pei Chun Chen. «A benchmark for machine-learning based non-invasive blood pressure estimation using photoplethysmogram». In: *Scientific Data 2023 10:1* 10 (1 Mar. 2023), pp. 1–16. ISSN: 2052-4463. DOI: 10.1038/s41597-023-02020-6. URL: https://www.nature.com/articles/s41597-023-02020-6 (cit. on pp. 3, 21, 30–32, 46).

[3] https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds). Accessed: 03-11-2024 (cit. on p. 4).

[4] https://www.who.int/news-room/fact-sheets/detail/hypertension. Accessed: 03-11-2024 (cit. on p. 4).

[5] https://www.cdc.gov/high-blood-pressure/about/index.html. Accessed: 03-11-2024 (cit. on p. 4).

[6] https://en.wikipedia.org/wiki/Vital_signs. Accessed: 03-11-2024 (cit. on p. 5).

[7] Yinji Ma et al. «Relation between blood pressure and pulse wave velocity for human arteries». eng. In: *Proceedings of the National Academy of Sciences of the United States of America* 115.44 (Oct. 2018), pp. 11144–11149. ISSN: 1091-6490. DOI: 10.1073/pnas.1814392115 (cit. on p. 10).

[8] Xiao-Rong Ding, Yuan-Ting Zhang, Jing Liu, Wen-Xuan Dai, and Hon Ki Tsang. «Continuous Cuffless Blood Pressure Estimation Using Pulse Transit Time and Photoplethysmogram Intensity Ratio». In: *IEEE Transactions on Biomedical Engineering* 63.5 (2016), pp. 964–972. DOI: 10.1109/TBME.2015.2480679 (cit. on p. 10).

[9] Alessio Burrello, Daniele Jahier Pagliari, Pierangelo Maria Rapa, Matilde Semilia, Matteo Risso, Tommaso Polonelli, Massimo Poncino, Luca Benini, and Simone Benatti. «Embedding Temporal Convolutional Networks for Energy-efficient PPG-based Heart Rate Monitoring». In: *ACM Transactions on Computing for Healthcare* 3.2 (Mar. 2022), 19:1–19:25. DOI: `10.1145/3487910`. URL: `https://dl.acm.org/doi/10.1145/3487910` (visited on 10/20/2023) (cit. on p. 12).

[10] Panagiotis Kasnesis, Lazaros Toumanidis, Alessio Burrello, Christos Chatzigeorgiou, and Charalampos Z. Patrikakis. «Feature-Level Cross-Attentional PPG and Motion Signal Fusion for Heart Rate Estimation». In: *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. ISSN: 0730-3157. June 2023, pp. 1731–1736. DOI: `10.1109/COMPSAC57700.2023.00267`. URL: `https://ieeexplore.ieee.org/abstract/document/10196998` (visited on 10/20/2023) (cit. on p. 12).

[11] Alessio Burrello, Daniele Jahier Pagliari, Marzia Bianco, Enrico Macii, Luca Benini, Massimo Poncino, and Simone Benatti. «Improving PPG-based Heart-Rate Monitoring with Synthetically Generated Data». In: *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. ISSN: 2163-4025. Oct. 2022, pp. 153–157. DOI: `10.1109/BioCAS54905.2022.9948584`. URL: `https://ieeexplore.ieee.org/abstract/document/9948584` (visited on 10/20/2023) (cit. on pp. 12, 33).

[12] `https://support.apple.com/en-us/108375`. Accessed: 30-10-2024 (cit. on p. 12).

[13] Jae-Hak Jeong, Bomi Lee, Junki Hong, Tae-Heon Yang, and Yong-Hwa Park. «Reproduction of human blood pressure waveform using physiology-based cardiovascular simulator». In: *Scientific Reports* 13.1 (May 2023), p. 7856. ISSN: 2045-2322. DOI: `10.1038/s41598-023-35055-1`. URL: `https://doi.org/10.1038/s41598-023-35055-1` (cit. on p. 13).

[14] Taha Sochi. «Flow of Navier-Stokes Fluids in Cylindrical Elastic Tubes». In: *Journal of Applied Fluid Mechanics* 8 (Apr. 2015), pp. 181–188. DOI: `10.18869/acadpub.jafm.67.221.22802` (cit. on p. 13).

[15] Luca Formaggia, Daniele Lamponi, and Alfio Quarteroni. «One-dimensional models for blood flow in arteries». In: *Journal of Engineering Mathematics* 47.3 (Dec. 2003), pp. 251–276. ISSN: 1573-2703. DOI: `10.1023/B:ENGI.0000007980.01347.29`. URL: `https://doi.org/10.1023/B:ENGI.0000007980.01347.29` (cit. on p. 13).

[16] «Wearable Photoplethysmography for Cardiovascular Monitoring». In: *Proceedings of the Ieee. Institute of Electrical and Electronics Engineers* 110.3 (Jan. 2022), pp. 355–381. ISSN: 0018-9219. DOI: `10.1109/JPROC.2022.3149785`. URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7612541/` (visited on 02/27/2024) (cit. on p. 13).

[17] José Guilherme Chaui-Berlinck and José Eduardo Pereira Wilken Bicudo. «The Scaling of Blood Pressure and Volume». In: *Foundations* 1.1 (2021), pp. 145–154. ISSN: 2673-9321. DOI: `10.3390/foundations1010010`. URL: `https://www.mdpi.com/2673-9321/1/1/10` (cit. on p. 14).

[18] Andrew Reisner, Phillip A. Shaltis, Devin McCombie, H Harry Asada, David S. Warner, and Mark A. Warner. «Utility of the Photoplethysmogram in Circulatory Monitoring». In: *Anesthesiology* 108.5 (May 2008), pp. 950–958. ISSN: 0003-3022. DOI: `10.1097/ALN.0b013e31816c89e1`. URL: `https://doi.org/10.1097/ALN.0b013e31816c89e1` (visited on 02/27/2024) (cit. on p. 14).

[19] Gloria Martínez, Newton Howard, Derek Abbott, Kenneth Lim, Rabab Ward, and Mohamed Elgendi. «Can Photoplethysmography Replace Arterial Blood Pressure in the Assessment of Blood Pressure?» In: *Journal of Clinical Medicine* 7.10 (2018). ISSN: 2077-0383. DOI: `10.3390/jcm7100316`. URL: `https://www.mdpi.com/2077-0383/7/10/316` (cit. on p. 14).

[20] Clare Bycroft et al. «The UK Biobank resource with deep phenotyping and genomic data». In: *Nature* 562.7726 (Oct. 2018), pp. 203–209. ISSN: 1476-4687. DOI: `10.1038/s41586-018-0579-z`. URL: `https://doi.org/10.1038/s41586-018-0579-z` (cit. on p. 16).

[21] Alistair E. W. Johnson et al. «MIMIC-IV, a freely accessible electronic health record dataset». In: *Scientific Data* 10.1 (Jan. 2023), p. 1. ISSN: 2052-4463. DOI: `10.1038/s41597-022-01899-x`. URL: `https://doi.org/10.1038/s41597-022-01899-x` (cit. on p. 16).

[22] M. Saeed, C. Lieu, G. Raber, and R.G. Mark. «MIMIC II: a massive temporal ICU patient database to support research in intelligent patient monitoring». In: *Computers in Cardiology*. 2002, pp. 641–644. DOI: `10.1109/CIC.2002.1166854` (cit. on p. 16).

[23] Alistair E.W. Johnson et al. «MIMIC-III, a freely accessible critical care database». In: *Scientific Data* 3.1 (May 2016), p. 160035. ISSN: 2052-4463. DOI: `10.1038/sdata.2016.35`. URL: `https://doi.org/10.1038/sdata.2016.35` (cit. on p. 16).

[24] Hyung-Chul Lee, Yoonsang Park, Soo Bin Yoon, Seong Mi Yang, Dongnyeok Park, and Chul-Woo Jung. «VitalDB, a high-fidelity multi-parameter vital signs database in surgical patients». In: *Scientific Data* 9.1 (June 2022), p. 279. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01411-5. URL: https://doi.org/10.1038/s41597-022-01411-5 (cit. on p. 16).

[25] Guillaume Weber-Boisvert, Benoit Gosselin, and Frida Sandberg. «Intensive care photoplethysmogram datasets and machine-learning for blood pressure estimation: Generalization not guarantied». In: *Frontiers in Physiology* 14 (2023). ISSN: 1664-042X. DOI: 10.3389/fphys.2023.1126957. URL: https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2023.1126957 (cit. on p. 17).

[26] Umit Senturk, Ibrahim Yucedag, and Kemal Polat. «Repetitive neural network (RNN) based blood pressure estimation using PPG and ECG signals». en. In: *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. Ankara: IEEE, Oct. 2018, pp. 1–4. ISBN: 978-1-5386-4184-2. DOI: 10.1109/ISMSIT.2018.8567071. URL: https://ieeexplore.ieee.org/document/8567071/ (visited on 09/29/2023) (cit. on p. 17).

[27] Sakib Mahmud et al. «A Shallow U-Net Architecture for Reliably Predicting Blood Pressure (BP) from Photoplethysmogram (PPG) and Electrocardiogram (ECG) Signals». en. In: *Sensors* 22.3 (Jan. 2022). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 919. ISSN: 1424-8220. DOI: 10.3390/s22030919. URL: https://www.mdpi.com/1424-8220/22/3/919 (visited on 10/03/2023) (cit. on p. 17).

[28] Jie Shan Vanessa Leong and Kok Beng Gan. «Cuffless Non-invasive Blood Pressure Measurement Using CNN-LSTM Model: A Correlation Study». en. In: *International Journal on Robotics, Automation and Sciences* 5.2 (Sept. 2023). Number: 2, pp. 25–32. ISSN: 2682-860X. DOI: 10.33093/ijoras.2023.5.2.3. URL: https://mmupress.com/index.php/ijoras/article/view/552 (visited on 10/03/2023) (cit. on p. 17).

[29] Richard Byfield, Morgan Miller, Jonathan Miles, Giovanna Guidoboni, and Jian Lin. «Towards Robust Blood Pressure Estimation From Pulse Wave Velocity Measured by Photoplethysmography Sensors». en. In: *IEEE Sensors Journal* 22.3 (Feb. 2022), pp. 2475–2483. ISSN: 1530-437X, 1558-1748, 2379-9153. DOI: 10.1109/JSEN.2021.3134890. URL: https://ieeexplore.ieee.org/document/9646921/ (visited on 09/29/2023) (cit. on p. 18).

[30] Moajjem Hossain Chowdhury, Md Nazmul Islam Shuzan, Muhammad E.H. Chowdhury, Zaid B. Mahbub, M. Monir Uddin, Amith Khandakar, and Mamun Bin Ibne Reaz. «Estimating Blood Pressure from the Photoplethysmogram Signal and Demographic Features Using Machine Learning Techniques». en. In: *Sensors* 20.11 (June 2020), p. 3127. ISSN: 1424-8220. DOI: `10.3390/s20113127`. URL: `https://www.mdpi.com/1424-8220/20/11/3127` (visited on 09/29/2023) (cit. on p. 18).

[31] Nabil Ibtehaz, Sakib Mahmud, Muhammad E. H. Chowdhury, Amith Khandakar, Muhammad Salman Khan, Mohamed Arselene Ayari, Anas M. Tahir, and M. Sohel Rahman. «PPG2ABP: Translating Photoplethysmogram (PPG) Signals to Arterial Blood Pressure (ABP) Waveforms». eng. In: *Bioengineering (Basel, Switzerland)* 9.11 (Nov. 2022), p. 692. ISSN: 2306-5354. DOI: `10.3390/bioengineering9110692` (cit. on p. 18).

[32] Qunfeng Tang, Zhencheng Chen, Rabab Ward, Carlo Menon, and Mohamed Elgendi. «Subject-Based Model for Reconstructing Arterial Blood Pressure from Photoplethysmogram». en. In: *Bioengineering* 9.8 (Aug. 2022), p. 402. ISSN: 2306-5354. DOI: `10.3390/bioengineering9080402`. URL: `https://www.mdpi.com/2306-5354/9/8/402` (visited on 03/22/2024) (cit. on pp. 18, 24).

[33] Annunziata Paviglianiti, Vincenzo Randazzo, Stefano Villata, Giansalvo Cirrincione, and Eros Pasero. «A Comparison of Deep Learning Techniques for Arterial Blood Pressure Prediction». In: *Cognitive Computation* 14.5 (Sept. 2022), pp. 1689–1710. ISSN: 1866-9964. DOI: `10.1007/s12559-021-09910-0`. URL: `https://doi.org/10.1007/s12559-021-09910-0` (cit. on p. 18).

[34] Yan Chu, Kaichen Tang, Yu-Chun Hsu, Tongtong Huang, Dulin Wang, Wentao Li, Sean I. Savitz, Xiaoqian Jiang, and Shayan Shams. «Non-invasive arterial blood pressure measurement and SpO2 estimation using PPG signal: a deep learning framework». en. In: *BMC Medical Informatics and Decision Making* 23.1 (Dec. 2023). Number: 1 Publisher: BioMed Central, pp. 1–16. ISSN: 1472-6947. DOI: `10.1186/s12911-023-02215-2`. URL: `https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-023-02215-2` (visited on 02/28/2024) (cit. on p. 19).

[35] Nicolas Aguirre, Edith Grall-Maës, Leandro J. Cymberknop, and Ricardo L. Armentano. «Blood Pressure Morphology Assessment from Photoplethysmogram and Demographic Information Using Deep Learning with Attention Mechanism». en. In: *Sensors* 21.6 (Jan. 2021). Number: 6 Publisher: Multidisciplinary Digital Publishing Institute, p. 2167. ISSN: 1424-8220. DOI: `10.3390/s21062167`. URL: `https://www.mdpi.com/1424-8220/21/6/2167` (visited on 10/03/2023) (cit. on pp. 19, 21).

[36] Jehyun Kyung, Joon-Young Yang, Jeong-Hwan Choi, Joon-Hyuk Chang, Sangkon Bae, Jinwoo Choi, and Younho Kim. «Deep-learning-based blood pressure estimation using multi channel photoplethysmogram and finger pressure with attention mechanism». en. In: *Scientific Reports* 13.1 (June 2023). Number: 1 Publisher: Nature Publishing Group, p. 9311. ISSN: 2045-2322. DOI: `10.1038/s41598-023-36068-6`. URL: `https://www.nature.com/articles/s41598-023-36068-6` (visited on 02/27/2024) (cit. on p. 19).

[37] Josep Sola, Anna Vybornova, Sibylle Fallet, Erietta Polychronopoulou, Arlene Wurzner-Ghajarzadeh, and Gregoire Wuerzner. «Validation of the optical Aktiia bracelet in different body positions for the persistent monitoring of blood pressure». en. In: *Scientific Reports* 11.1 (Oct. 2021), p. 20644. ISSN: 2045-2322. DOI: `10.1038/s41598-021-99294-w`. URL: `https://www.nature.com/articles/s41598-021-99294-w` (visited on 09/29/2023) (cit. on p. 19).

[38] Lindercy Francisco Tomé de Souza Lins, Ellany Gurgel Cosme do Nascimento, José Antonio da Silva Júnior, Thales Allyrio Araújo de Medeiros Fernandes, Micássio Fernandes de Andrade, and Cléber de Mesquita Andrade. «Accuracy of wearable electronic device compared to manual and automatic methods of blood pressure determination». eng. In: *Medical & Biological Engineering & Computing* 61.10 (Oct. 2023), pp. 2627–2636. ISSN: 1741-0444. DOI: `10.1007/s11517-023-02869-0` (cit. on p. 20).

[39] Patrick Schoettker et al. «Blood pressure measurements with the OptiBP smartphone app validated against reference auscultatory measurements». In: *Scientific Reports* 10.1 (Oct. 2020), p. 17827. ISSN: 2045-2322. DOI: `10.1038/s41598-020-74955-4`. URL: `https://doi.org/10.1038/s41598-020-74955-4` (cit. on p. 20).

[40] Edward Jay Wang, Junyi Zhu, Mohit Jain, Tien-Jui Lee, Elliot Saba, Lama Nachman, and Shwetak N. Patel. «Seismo: Blood Pressure Monitoring using Built-in Smartphone Accelerometer and Camera». In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. New York, NY, USA: Association for Computing Machinery, Apr. 2018, pp. 1–9. ISBN: 978-1-4503-5620-6. DOI: `10.1145/3173574.3173999`. URL: `https://doi.org/10.1145/3173574.3173999` (visited on 08/08/2024) (cit. on p. 20).

[41] Keke Qin, Wu Huang, Tao Zhang, and Shiqi Tang. «Machine learning and deep learning for blood pressure prediction: a methodological review from multiple perspectives». In: *Artificial Intelligence Review* 56.8 (Dec. 2022), pp. 8095–8196. ISSN: 0269-2821. DOI: `10.1007/s10462-022-10353-8`. URL: `https://doi.org/10.1007/s10462-022-10353-8` (visited on 02/28/2024) (cit. on p. 20).

[42] Nicolas Aguirre, Edith Grall-Maës, Leandro Javier Cymberknop, and Ricardo Luis Armentano. *Dataset corresponding to "Blood Pressure Morphology Assessment from Photoplethysmogram and Demographic Information Using Deep Learning with Attention Mechanism"*. Version 1.0. Zenodo, Mar. 2021. DOI: 10.5281/zenodo.4598938. URL: https://doi.org/10.5281/zenodo.4598938 (cit. on p. 21).

[43] Mohamad Kachuee, Mohammad Mahdi Kiani, Hoda Mohammadzade, and Mahdi Shabany. «Cuff-less high-accuracy calibration-free blood pressure estimation using pulse transit time». In: *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. ISSN: 2158-1525. May 2015, pp. 1006–1009. DOI: 10.1109/ISCAS.2015.7168806. URL: https://ieeexplore.ieee.org/document/7168806 (visited on 08/12/2024) (cit. on p. 21).

[44] Charles Carlson, Vanessa-Rose Turpin, Ahmad Suliman, Carl Ade, Steve Warren, and David E. Thompson. «Bed-Based Ballistocardiography: Dataset and Ability to Track Cardiovascular Parameters». In: *Sensors* 21.1 (2021). ISSN: 1424-8220. DOI: 10.3390/s21010156. URL: https://www.mdpi.com/1424-8220/21/1/156 (cit. on pp. 21, 22).

[45] Yongbo Liang, Zhencheng Chen, Guiyong Liu, and Mohamed Elgendi. «A new, short-recorded photoplethysmogram dataset for blood pressure monitoring in China». en. In: *Scientific Data* 5.1 (Feb. 2018), p. 180020. ISSN: 2052-4463. DOI: 10.1038/sdata.2018.20. URL: https://www.nature.com/articles/sdata201820 (visited on 09/29/2023) (cit. on pp. 21, 22).

[46] Mohamad Kachuee, Mohammad Kiani, Hoda Mohammadzade, and Mahdi Shabany. *Cuff-Less Blood Pressure Estimation*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5B602. 2015 (cit. on p. 21).

[47] Charles Carlson, Vanessa-Rose Turpin, Ahmad Suliman, Carl Ade, Steve Warren, and David E. Thompson. *Bed-Based Ballistocardiography Dataset*. 2020. DOI: 10.21227/77hc-py84. URL: https://dx.doi.org/10.21227/77hc-py84 (cit. on p. 22).

[48] Yongbo Liang, Guiyong Liu, Zhencheng Chen, and Mohamed Elgendi. *PPG-BP Database*. Feb. 2018. DOI: 10.6084/m9.figshare.5459299.v5. URL: https://figshare.com/articles/dataset/PPG-BP_Database_zip/5459299 (cit. on p. 22).

[49] Hsieh Wan-Ting, Vázquez Sergio González, and Chen Trista. *A benchmark for machine-learning based non-invasive blood pressure estimation using photoplethysmogram*. 2023. DOI: https://doi.org/10.6084/m9.figshare.c.6150390.v1. URL: https://doi.org/10.6084/m9.figshare.c.6150390.v1 (cit. on p. 22).

[50] https://github.com/inventec-ai-center/bp-benchmark. Accessed: 15-8-2024 (cit. on p. 22).

[51] Lida Zhang, Nathan C. Hurley, Bassem Ibrahim, Erica Spatz, Harlan M. Krumholz, Roozbeh Jafari, and Bobak J. Mortazavi. «Developing Personalized Models of Blood Pressure Estimation from Wearable Sensors Data Using Minimally-trained Domain Adversarial Neural Networks». In: *Proceedings of machine learning research* 126 (Aug. 2020), pp. 97–120. ISSN: 2640-3498. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7916101/ (visited on 07/30/2024) (cit. on p. 24).

[52] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. «Designing Neural Network Architectures using Reinforcement Learning». In: *ArXiv* abs/1611.02167 (2016). URL: https://api.semanticscholar.org/CorpusID:1740355 (cit. on p. 25).

[53] Joseph Charles Mellor, Jack Turner, Amos J. Storkey, and Elliot J. Crowley. «Neural Architecture Search without Training». In: *ArXiv* abs/2006.04647 (2020). URL: https://api.semanticscholar.org/CorpusID:219531078 (cit. on p. 25).

[54] Meng-Ting Wu and Chun-Wei Tsai. «Training-free neural architecture search: A review». In: *ICT Express* 10.1 (2024), pp. 213–231. ISSN: 2405-9595. DOI: https://doi.org/10.1016/j.icte.2023.11.001. URL: https://www.sciencedirect.com/science/article/pii/S2405959523001443 (cit. on p. 25).

[55] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. «Efficient Neural Architecture Search via Parameter Sharing». In: *ArXiv* abs/1802.03268 (2018). URL: https://api.semanticscholar.org/CorpusID:3638969 (cit. on p. 25).

[56] Hanxiao Liu, Karen Simonyan, and Yiming Yang. «DARTS: Differentiable Architecture Search». In: *arXiv preprint arXiv:1806.09055* (2018) (cit. on p. 25).

[57] Matteo Risso, Alessio Burrello, Daniele Jahier Pagliari, Francesco Conti, Lorenzo Lamberti, Enrico Macii, Luca Benini, and Massimo Poncino. «Pruning In Time (PIT): A Lightweight Network Architecture Optimizer for Temporal Convolutional Networks». en. In: *2021 58th ACM/IEEE Design Automation Conference (DAC)*. arXiv:2203.14768 [cs]. Dec. 2021, pp. 1015–1020. DOI: 10.1109/DAC18074.2021.9586187. URL: http://arxiv.org/abs/2203.14768 (visited on 11/03/2023) (cit. on p. 26).

[58] Matteo Risso, Alessio Burrello, Luca Benini, Enrico Macii, Massimo Poncino, and Daniele Jahier Pagliari. «Channel-wise Mixed-precision Assignment for DNN Inference on Constrained Edge Nodes». In: *2022 IEEE 13th International Green and Sustainable Computing Conference (IGSC)*. 2022, pp. 1–6. DOI: 10.1109/IGSC55832.2022.9969373 (cit. on p. 26).

[59] Alessio Burrello, Francesco Carlucci, Giovanni Pollo, Xiaying Wang, Massimo Poncino, Enrico Macii, Luca Benini, and Daniele Jahier Pagliari. *Optimization and Deployment of Deep Neural Networks for PPG-based Blood Pressure Estimation Targeting Low-power Wearables.* 2024. arXiv: 2409.07485 [eess.SP]. URL: https://arxiv.org/abs/2409.07485 (cit. on p. 27).

[60] Matteo Risso, Alessio Burrello, Francesco Conti, Lorenzo Lamberti, Yukai Chen, Luca Benini, Enrico Macii, Massimo Poncino, and Daniele Jahier Pagliari. «Lightweight Neural Architecture Search for Temporal Convolutional Networks at the Edge». In: *IEEE Transactions on Computers* 72.3 (2023), pp. 744–758. DOI: 10.1109/TC.2022.3177955 (cit. on pp. 27, 28).

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. «Deep residual learning for image recognition». In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 770–778 (cit. on p. 29).

[62] Jie Hu, Li Shen, and Gang Sun. «Squeeze-and-Excitation Networks». In: 2018 (cit. on p. 30).

[63] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. «U-net: Convolutional networks for biomedical image segmentation». In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18.* Springer. 2015, pp. 234–241 (cit. on p. 31).

[64] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. «MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications». In: *CoRR* abs/1704.04861 (2017). arXiv: 1704.04861. URL: http://arxiv.org/abs/1704.04861 (cit. on p. 38).

[65] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. «PACT: Parameterized Clipping Activation for Quantized Neural Networks». In: *CoRR* abs/1805.06085 (2018). arXiv: 1805.06085. URL: http://arxiv.org/abs/1805.06085 (cit. on p. 39).

[66] Angelo Garofalo, Manuele Rusci, Francesco Conti, Davide Rossi, and Luca Benini. «PULP-NN: Accelerating Quantized Neural Networks on Parallel Ultra-Low-Power RISC-V Processors». In: *CoRR* abs/1908.11263 (2019). arXiv: `1908.11263`. URL: `http://arxiv.org/abs/1908.11263` (cit. on p. 39).