# Separation of Vibrational Cardiography Signals by Respiratory Volume and Phase using 1-Dimensional Convolutional Neural Networks

*Nathan Clairmonte*

Department of Electrical & Computer Engineering

McGill University

Montreal, Canada

January 2023

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Science

# ABSTRACT

Cardiovascular disease has been the leading cause of human mortality globally for years. Increasing cardiovascular health could reduce the frequency of cardiovascular disease and save lives, and preventative care has the potential to increase cardiovascular health. However, preventative care for cardiovascular disease is not ideal, most importantly lacking an established non-invasive, continuous method of cardiac monitoring. Non-invasive health monitoring could improve the application and efficiency of medical treatment and significantly reduce the frequency of cardiovascular disease-related mortality rates. Vibrational cardiography (VCG) has the potential to deliver non-invasive cardio-respiratory monitoring. VCG is the term given to a coupled seismocardiography (SCG) and gyrocardiography (GCG) measurement. VCG (along with its components, SCG and GCG) have been well studied and developed for cardiac monitoring. Moreover, the inherent effects of respiration on the VCG signal due to the proximity of the lungs to the heart have been studied as well. However, there is no established method of mitigating the respiratory variation in a VCG signal, thus reducing its efficacy as a cardiac monitoring tool. Approaches have been taken to filter out respiratory information from the VCG signal entirely, but studies have shown that this respiratory information could be useful for monitoring cardiovascular health. Instead, other approaches have been taken to separate VCG signals based on the respiratory phase or volume of the subject at the time they were recorded. This reduces respiratory variation in the signal without losing the potentially useful respiratory information altogether. The objective of this thesis is to take this separation approach, classifying VCG signals based on their respiratory volume and phase, specifically using 1-dimensional (1D) convolutional neural networks (CNN). 1D CNNs are artificial neural networks which apply convolving filters to local features in one dimension. These networks are especially useful for analysing data in the temporal dimension and have been shown to have excellent performance in many signal processing domains, hence why they were chosen for this analysis. Data were collected from 50 subjects at McGill University, using an inertial measurement unit taped to the chest to obtain a VCG signal, and a spirometer to obtain a reference respiratory flow signal. Three classification objectives were examined: static respiratory volume, dynamic respiratory volume, and dynamic respiratory phase. For each objective, the cardiac cycles obtained from the VCG signals were manually split into one of two classes based on the respiratory flow signal and a 1D CNN was employed to classify these cardiac cycles based solely on their VCG information.

Subsequently, experiments were performed on the CNN architecture to determine the optimal architecture for each of the classification objectives. It was found that a 1D CNN has the ability to classify VCG signals based on the respiratory volume and phase of the subject at the time they were recorded, achieving harmonic mean of precision and recall (F1) scores of 0.9972, 0.8324 and 0.8617 for the static respiratory volume, dynamic respiratory volume and dynamic respiratory phase classification objectives, respectively. Moreover, it was found that in many cases these scores were improved by reducing the complexity of the models used. Our findings show that 1D CNNs are a viable method to mitigate respiratory variation in VCG signals. Moreover, the finding that less complex models outperformed more complex models is significant because it increases the viability of deploying such models in low-power, low-memory or mobile applications. More work should be explored as to why less complex models tended to perform better, and a more detailed study on a larger population to validate these findings would be beneficial. VCG is a powerful tool that has the potential to provide non-invasive health monitoring for both cardiac and respiratory applications.

# RÉSUMÉ

La maladie cardiovasculaire est la cause majeure de la mortalité humaine dans le monde depuis des années. L'amélioration de la santé cardiovasculaire peut réduire son incidence tout en sauvant la vie. Les soins préventifs ont le potentiel d'améliorer la santé cardiovasculaire. Pourtant, les soins destinés à prévenir la maladie cardiovasculaire ne constituent pas une solution idéale dans la mesure où ils manquent une méthode systématique de surveillance cardiaque non invasive. La surveillance sanitaire non invasive peut améliorer l'application et l'efficacité des soins médicaux tout en réduisant la fréquence du taux de mortalité relatif à la maladie cardiovasculaire: La cardiographie vibrationnelle (VCG) a le potentiel de fournir la surveillance non invasive cardio-respiratoire. La VCG désigne la mesure qui combine la seismocardiographie (SCG) et la gyrocardiographie (GCG). La VCG (y compris ses composantes, la SCG et la GCG) relatives à la surveillance cardiaque ont été bien étudiées et mises au point. En outre, les effets inhérents de la respiration sur le signal VCG en raison de la proximité des poumons au coeur, ont également été étudiés. Pourtant, il n'existe pas de méthode établie pour minimiser la variation respiratoire présente dans le signal VCG, réduisant donc son efficacité d'outil de la surveillance cardiaque. Certaines approches ont été adoptées pour éliminer complètement les données respiratoires du signal VCG, mais des études ont démontré que ces données peuvent être utiles pour suivre la santé cardiovasculaire. A leur place, d'autres méthodes ont été adoptées pour séparer les signaux VCG basées sur la phase respiratoire ou volume du sujet au moment de l'enregistrement. Cette méthode minimise la variation respiratoire du signal, sans perdre complètement l'utilité éventuelle au niveau de la respiration. Cette thèse a comme objectif l'adoption de cette méthode de séparation, tout en classant les signaux VCG basés sur leur phase et volume respiratoire, utilisant spécifiquement des réseaux neuronaux convolutifs (RNC) unidimensionnels (UD). Les RNC UD sont des réseaux neuronaux artificiels qui appliquent des filtres convolutifs aux éléments locaux dans une seule dimension. Ces réseaux sont particulièrement utiles pour analyser les données en dimension temporelle et ont démontré une performance excellente dans de nombreux domaines de traitement du signal, ce qui explique le choix de sujet de cette analyse. Les données ont été recueillies auprès d'un échantillon de 50 personnes (les sujets) à l'Université de McGill, utilisant une unité de mesure inertielle attachée à la poitrine pour obtenir un signal VCG, ainsi qu'un spiromètre en vue d' obtenir un signal de débit respiratoire. Trois objectifs de classification ont été examinés: volume respiratoire statique,

volume respiratoire dynamique, et phase respiratoire dynamique. Pour chaque objectif, les cycles cardiaques obtenus auprès des signaux VCG ont été manuellement divisés en une de deux classes basés sur le signal de débit respiratoire et un RNC UD a été utilisé pour classer ces cycles cardiaques basés uniquement sur les données émanant de leur VCG. Par la suite, des expérimentations ont été menées sur l'architecture RNC en vue d'établir l'architecture optimale pour chacun des objectifs de classification. Il a été constaté qu' un RNC UN est capable de classer des signaux VCG basé sur le signal de débit respiratoire et la phase du sujet au moment de l'enregistrement, atteignant respectivement des scores F1 de 0,9972, 0,8324 et 0,8617 pour le volume respiratoire statique, volume respiratoire dynamique, et phase respiratoire dynamique des objectifs de classification. En outre, il a été constaté que, dans de nombreux cas, ces scores ont été améliorés en minimisant la complexité des modèles utilisés. Nos résultats démontrent que des RNC UD constituent une méthode efficace pour minimiser la variation respiratoire des signaux VCG. En outre, le résultat que les modèles moins complexes ont surpassé les modèles plus complexes est important dans la mesure où cela augmente la viabilité du lancement de tels modèles dans des applications à faible-puissance, à faible-mémoire ou portable. Des recherches plus approfondies doivent être menées afin d'expliquer pourquoi des modèles moins complexes ont la tendance de mieux fonctionner, et une étude plus minutieuse sur un plus grand échantillon pour valider ces résultats serait utile. La VCG est un outil puissant possédant le potentiel de fournir à la fois la surveillance sanitaire non invasive à l'intention des applications cardiaques et respiratoires.

# ACKNOWLEDGEMENTS

I would like to firstly thank my supervisor, Professor David V. Plant, for giving me the opportunity to conduct research I am so passionate about in his research group, and for his endless advice, leadership and assistance throughout the course of my research. His belief in me and his constant support has been crucial to this endeavour.

I would especially like to express my gratitude to my two biggest and most influential collaborators, Yannick D'Mello and James Skoric, whose guidance from the time I was an undergraduate student led me towards this research and helped me at every point along the way with it. They were my introduction to the world of research and two of the biggest reasons I fell in love with this project. They taught me how to think like a researcher and how to navigate the often-difficult path of a master's student. Without their expert advice, assistance and sometimes criticism, this project would not be what it is today. I cannot express enough gratitude to these brilliant researchers for everything they have done for me. They began their journey with me as colleagues, and they have grown to become friends whom I will cherish forever.

I would like to thank all of the other members of the Non-invasive Physical Activity Monitoring System (NiPAMS) team for their help in achieving my goals throughout this project. Namely, Ezz Aboulezz for his assistance with the acquisition system design and setup, Siddiqui Hakim and Angus McLean for their assistance with the data acquisition for the project, and Michel Lortie of the MDA corporation for supporting and trusting in this project throughout.

I would like to thank my friends, both back home and in Canada, for being my support system throughout the course of this research. There are too many to name them all, but I would like to give special thanks to Sebastian Hunte, Brian Wood, Jaec Emtage-Cave, Isidora Conic and Stuart St. Hill for taking the time to review my work and help me streamline my phrasings and explanations to make sure they could be understood by those not familiar with the project.

Finally, and perhaps most importantly, I would like to thank my parents, Michele Clairmonte and Frederick Clairmonte for their constant support throughout my entire life. From encouraging me to pursue my interests in math and physics as a child, to supporting my decision to pursue undergraduate and master's degrees overseas, to encouraging me and being there for me during the lowest points in my life. They have been absolutely crucial to every single one of

my successes and words cannot express how much I owe them for everything I have today. Thank you.

# AUTHOR CONTRIBUTIONS

This thesis in its entirety was written by Nathan Clairmonte. All chapters of this thesis represent work completed by myself. Specifically, I contributed to the experimental setup design, human trial designs, data collection, data processing, feature construction, model selection and curation, model training, model optimization and creation of all images in this work. James Skoric, Yannick D'Mello and Ezz Aboulezz contributed to the experimental setup design. James Skoric and Yannick D'Mello contributed to the human trial designs and overall project framework. Ezz Aboulezz, Siddiqui Hakim and Angus McLean contributed to the data acquisition. French translation of the thesis abstract was done by Mr. Martin Wood and Mrs. Melba Wood.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ACRONYMS

| | |
|---|---|
| **1D** | 1-Dimensional |
| **2D** | 2-Dimensional |
| **ANN** | Artificial Neural Networks |
| **BCG** | Ballistocardiography |
| **CNN** | Convolutional Neural Network |
| **CVD** | Cardiovascular Disease |
| **CVH** | Cardiovascular Health |
| **ECG** | Electrocardiography |
| **FN** | False Negative |
| **FP** | False Positive |
| **FTP** | File Transfer Protocol |
| **GCG** | Gyrocardiography |
| **HLV** | High Lung Volume |
| **I2C** | Inter-Integrated Circuit |
| **ICG** | Impedance Cardiography |
| **IMU** | Inertial Measurement Unit |
| **LED** | Light-Emitting Diode |
| **LLV** | Low Lung Volume |
| **LSTM** | Long Short-Term Memory |
| **MEMS** | Micro-Electro-Mechanical Systems |
| **ML** | Machine Learning |
| **MLP** | Multi-Layer Perceptron |
| **MRI** | Magnetic Resonance Imaging |
| **PPG** | Photoplethysmography |
| **RIP** | Respiration Inductance Plethysmography |
| **RMS** | Root Mean Square |
| **RNN** | Recurrent Neural Networks |

| | |
|---|---|
| **ReLU** | Rectified Linear Unit |
| **SCG** | Seismocardiography |
| **SVM** | Support Vector Machine |
| **SVR** | Support Vector Regression |
| **TN** | True Negative |
| **TP** | True Positive |
| **VCG** | Vibrational Cardiography |

# I. INTRODUCTION

## A. Motivation

Cardiovascular disease (CVD) continues to be the leading cause of death across the world [1]. In 2020 alone, nearly 19 million deaths were attributed to CVD globally, amounting to an increase of 18.7% from 2010 [2]. In addition, an estimated 70.0% of the major instances of CVD events in the United States can be attributed to low and moderate cardiovascular health (CVH) [2]. Even a partial CVH score improvement from low to moderate among US adults could lead to a reduction of 1.2 million major CVD events annually [2]. The pervasiveness of CVD events and the global health benefits of increasing CVH, coupled with the fact that preventative care has the potential to reduce mortality rates by millions and economic losses by trillions [3], has incited the medical community to seek preventative measures in tackling CVD and improving CVH.

Preventative care for CVD often consists of regular check-ups with a physician, usually involving an assessment of CVH using one or more of the many cardiac and respiratory monitoring techniques available for a medical setting. However, uncommon or intermittent symptoms can easily be overlooked if they do not appear during the (usually brief) monitoring session [4]. Continuous monitoring options such as insertable cardiac monitors do exist, but their costs are significant; around $4,000 for the initial insertion and device in addition to subsequent monitoring costs. They also require an invasive procedure which can lead to rare complications [5]. As a result, these techniques are usually reserved for patients who are at a higher risk of atrial fibrillation due to existing cardiovascular complications such as a stroke [6]. Therefore, patients who do not fall into this category often do not receive continuous monitoring, or any cardio-respiratory monitoring outside of their regular check-ups. This demonstrates the need for an established non-invasive, continuous method of cardio-respiratory monitoring that remains accurate while not significantly interfering with daily life.

Non-invasive, continuous health monitoring could accelerate diagnoses, improve preventative care, and save lives by capitalising on algorithms that connect physiological signals to cardiovascular health state trajectories, and the potential of machine learning (ML) algorithms to identify such trends is evident [7]. This research used ML to investigate vibrational cardiography (VCG) as a possible solution.

## B. Non-invasive Cardio-respiratory Monitoring

### *1. Respiratory Monitoring*

Respiratory monitoring systems typically measure either airflow or chest movement to get a measure of respiratory activity. The most accurate method currently is to measure airflow and air volume using a spirometer. Commonly, spirometers measure air volume indirectly by measuring airflow with a pneumotachograph, turbines or ultrasound and deriving volume from it [8]. A pneumotachograph measures airflow by measuring the pressure change across a thin film inside the breathing tube which is linearly related to volumetric flow rate [9]. Usually, a pneumotachograph requires daily calibration [10], and heating at a constant temperature to mitigate pressure variances and condensation in the system. Additionally, it requires that the subject be breathing directly into the breathing tube continuously, which can be uncomfortable for the patient. Overall, these systems lack portability, require consistent calibration and re-calibration, and require the patient remain still while measurements are taken [11].

Another widely used wearable method is respiration inductance plethysmography (RIP). RIP measures the change in inductance of a coil within a band placed around the chest and uses this measurement to monitor torso expansion. Usually, two bands are utilised in tandem, with one monitoring abdomen expansion while the other monitors ribcage expansion. These systems can give a very accurate respiration rate and volume representation when calibrated properly [12]. However, their volumetric measurements are sensitive to changes in patient position or posture [13]. As a result, RIP has very limited portability because positional and postural changes during a measurement can result in large artifacts in the signal.

Alternatively, a nasal thermistor can be used to monitor airflow with a more portable system. A thermistor is a device whose electrical resistance changes with temperature. When fixed to a patient's face between their nose and mouth, this effect can be utilised to measure the temperature drop caused by the movement of air over the sensor with each breath. However, since the airflow from the patient's breath is not constrained, there is too much uncertainty for these systems to measure airflow or air volume. Therefore, they are most often used to estimate respiration rate and not flow [14]. In fact, their exact timing when compared to other methods has been disputed [15].

## *2. Cardiac Monitoring*

The current standard of non-invasively measuring cardiac activity is the electrocardiogram (ECG). ECG was first discovered towards the end of the 19th century, and its clinical usage began around the 1950's [16]. To obtain an ECG measurement, several electrodes are placed on a patient's torso, and they are used to measure the voltage change caused by the electrical activity of the heart [17]. This method is robust but can be complex for an inexperienced user, as its setup utilises a 12-lead configuration which requires the placement of 10 electrodes across the patient's chest in specific locations [17]. There is also a more portable ECG method called the Holter monitor. This method was discovered by Dr. Norman J. Holter in 1957 and is specifically designed for long-term, continuous ECG recording outside of a hospital setting. While Holter monitors can still utilise the aforementioned 12-lead configuration, they are also available in less complex 2-lead and 3-lead configurations which use significantly fewer electrodes [18]. ECG is commonly used to monitor heart rate [19]. It has also been shown to detect CVH-related abnormalities such as arrythmias [20] or ischemia [21]. It has even been shown to have reasonable accuracy in the prediction of overall mortality [22].

Impedance cardiography (ICG) is another method which is somewhat similar to ECG. However, instead of measuring changes in voltage, this method measures changes in electrical conductivity. The electrode placement for ICG also differs from that for ECG as it requires paired electrodes which are usually placed on the patient's neck and thorax [23]. Although this method is less complicated as the 12-lead ECG configuration and about as complicated as the 2-lead and 3-lead Holter monitor configurations, it is still not ideal for continuous monitoring as the placement of the electrodes on the neck can cause significant motion artifacts when patients move [24].

Photoplethysmography (PPG) is another method which has gained traction recently due to its portability and affordability. It has been utilised in many consumer electronics applications such as FitBits or Apple Watches for health and fitness monitoring. PPG operates by using a light-emitting diode (LED) and a photodetector to measure variations in blood circulation. Generally, the device is placed on the fingertip or earlobe. The LED emits light, and the photodetector measures the change in light absorption caused by blood flow [25]. Despite its portability and affordability, PPG still has drawbacks. The main one is that since PPG is

sensitive to ambient light, slight movement by the patient relative to light sources in the recording area can result in significant motion artifacts [26]. Generally, PPG is used as a replacement for ECG to monitor heart rate [27]. However, it has also been shown to provide information on oxygen saturation [25], cardiac abnormalities [28] or artery stiffness [29].

Other cardiac monitoring methods are available but are mostly limited in their continuous non-invasive monitoring applications due to the technical knowledge required for operation or a lack of portability. For example, echocardiography images the heart using ultrasound technology, but requires a trained operator to obtain accurate measurements [30]. Similarly, magnetic resonance imaging (MRI) technology can be used to image the heart but also requires a trained operator and lacks portability due to the large, expensive equipment required [31]. Even high-speed cameras have been utilised recently to detect heart rate by measuring the fluctuations in skin colour caused by blood circulation, but this method lacks portability [32]. Each of these methods have their own specific use-cases and their importance for cardiac monitoring should not be understated. However, for continuous, non-invasive and wearable applications, researchers are limited to simpler, more portable methods such as ECG and PPG. Another option for such applications is VCG.

## *3. Mechanical Cardiac Monitoring*

### ***a) Evolution***

Cardio-respiratory activity generates thoracic vibrations that propagate throughout the body [33]. These vibrations can easily be recorded non-invasively. This observation was first made as far back as the late 19$^{th}$ century [34]. In this work, subjects were placed on a bed suspended by ropes from the ceiling. The movements of the body imparted by the ballistic forces associated with cardio-respiratory activity and blood flow were recorded and found to be synchronous with the heartbeat. Subsequently, further work was done by many researchers, culminating in the emergence of ballistocardiography (BCG) in 1938 [35]. As BCG evolved, so did the associated vibrational recording techniques. For example, instead of a bed suspended by ropes, other researchers have utilised electrical apparatus such as the hot wire microphone. The hot wire microphone was originally designed in war times to detect enemy guns by measuring the pressure changes caused by the vibrations which emanated from the guns [36]. However, its

ability to detect vibrations made it an ideal candidate for detecting the vibrations in the body generated by cardio-respiratory activity [37].

***b) Seismocardiography***

With the advent of micro-electro-mechanical systems (MEMS)-based motion tracking technology, it became feasible to record these vibrations with a miniaturized accelerometer attached to the skin at the xiphoid process of the sternum (where vibrational signals are strongest due to the position of the heart in the thorax [38]). The recorded accelerometer signal is called a seismocardiography (SCG) signal [39].

One significant benefit of SCG is that since an accelerometer is used, the motion recorded is not solely constrained to cardiac activity – respiratory motion is also incorporated into the recording [40]. Since cardiac and respiratory monitoring techniques have largely evolved separately from each other, cardiac monitoring techniques often carry little respiratory information and vice versa. ECG and PPG have been shown to have slight respiratory effects in their signals, but this effect varies strongly based on the subject [41]. Moreover, respiratory monitoring techniques generally have no insight into cardiac activity whatsoever. This is not ideal, as cardiac and respiratory functions are inherently coupled and usually need to be monitored simultaneously. Therefore, the ability of SCG to capture both cardiac and respiratory activity in one signal and with one device is beneficial.

Furthermore, SCG has shown considerable accuracy in predicting several important cardiac time intervals such as the opening and closing of both the mitral and aortic valves [42]. During a single cardiac cycle, the mitral and aortic valves open and close as the heart contracts and expands. These valvular movements create specific vibrations during the cardiac cycle that appear on and can be defined as fiducial points on an SCG signal. An example of an SCG cardiac cycle with the fiducial points from [43] is shown in Figure 1. Since this work was one of the first to define SCG fiducial points, the naming conventions used have largely remained and are the current standard. However, some of their timing correlations have been both disputed [44] and renamed [45]. Using these fiducial points to track the timing of cardiac events has led SCG to emerge as a strong candidate for cardiac monitoring [38, 46]. In fact, cardiac detection using SCG has demonstrated accuracy at the same level as that of ECG [47]. This candidacy for cardiac monitoring is further strengthened by the fact that SCG sensors are

generally unobtrusive because they are small and only require a single point of contact on the body [48].

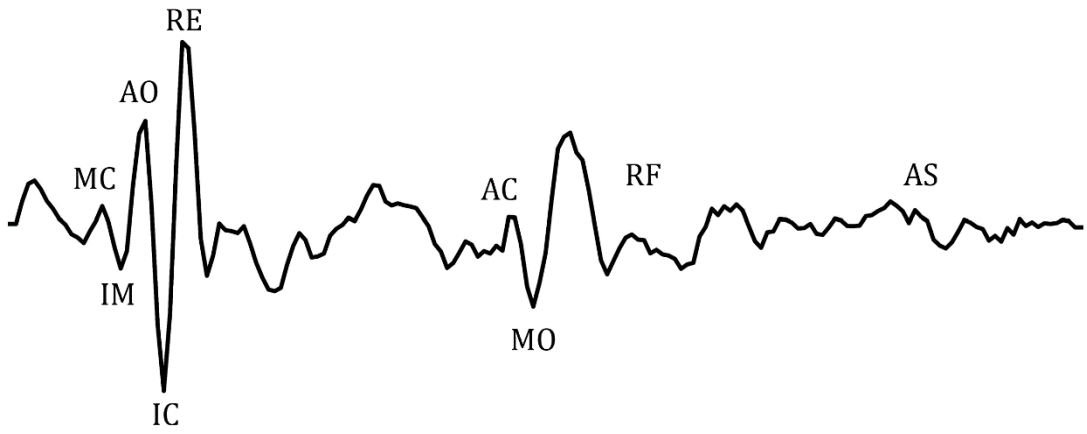


**Figure 1.** SCG cardiac cycle with fiducial points annotated. Fiducial points shown are the mitral valve closure (MC), isovolumic movement (IM), aortic valve opening (AO), isotonic contraction (IC), peak of rapid systolic ejection (RE), aortic valve closure (AC), mitral valve opening (MO), peak of rapid diastolic filling (RF), and peak of atrial systole (AS). Recreated using experimental data and fiducial points from [43].

Additionally, SCG can monitor respiration using similar principles to those of RIP – by monitoring the motion of the chest [49]. Many recent studies have also applied these principles, using cameras [50], wearable strain sensors [51], ultrasound [52] and radar [53] to monitor respiration via chest motion. However, outside of the high-speed cameras, these methods lack the granularity required to also record cardiac activity via chest motion. As a result, SCG remains one of the few current tools available that can monitor both cardiac and respiratory activity simultaneously.

***c) Vibrational Cardiography***

Recently, MEMS-based motion tracking technology has improved to the point where both an accelerometer and a gyroscope could be integrated into a single miniaturized inertial measurement unit (IMU). This provided an integrated, coupled gyration signal, which motivated research into gyrocardiography (GCG) as a complementary measurement [54]. In fact, over 50% of the total kinetic energy transferred from the heart to the body is contained in the GCG signal [55]. This shows the added value of incorporating GCG into cardio-respiratory monitoring

applications. Additionally, since the SCG and GCG measurements are mutually orthogonal by nature, there is an inherent difference in the noise characteristics to which each measurement is vulnerable. As a result, combining information from both measurements facilitates a more comprehensive analysis. This coupled SCG and GCG measurement has been termed vibrational cardiography (VCG) [47], and is the measurement utilised in this research. The signal morphologies for the six axes which constitute a single VCG cardiac cycle are shown in Figure **2**, with Figure **2**(a) showing the SCG axis components and Figure **2**(b) showing the GCG axis components.

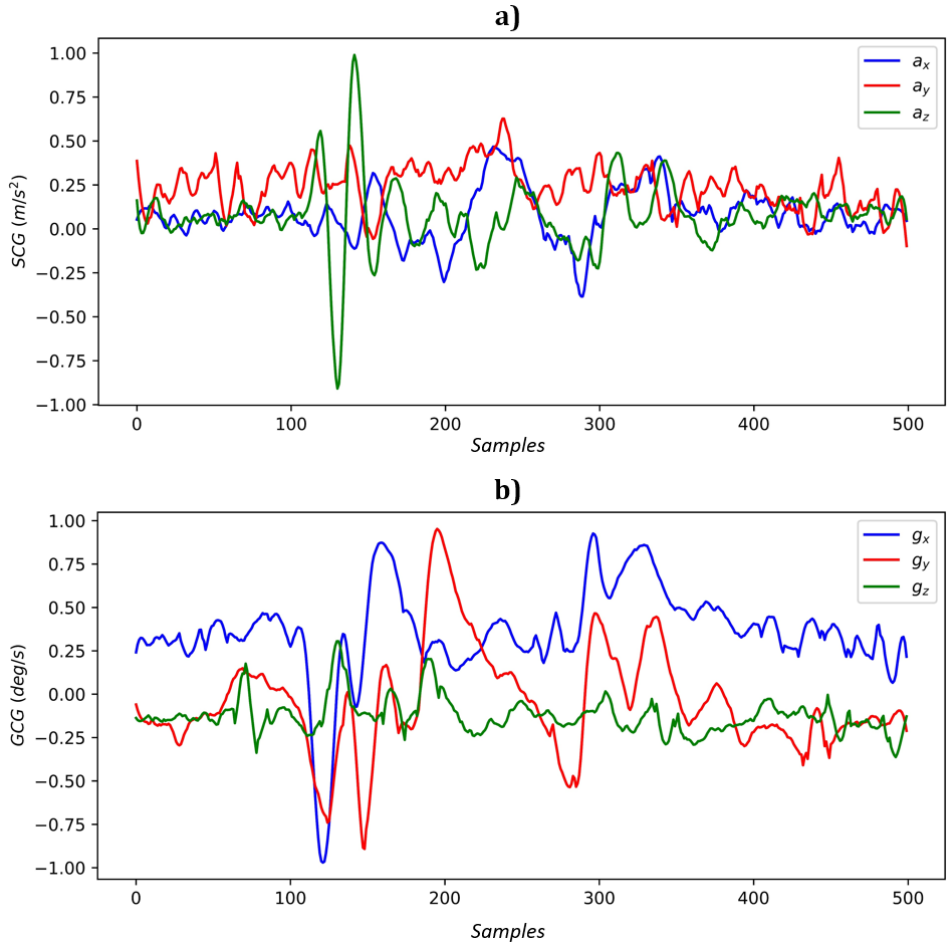


**Figure 2.** (a) Signal morphology of a single VCG cardiac cycle shown for acceleration in all three axis components. (b) Signal morphology of a single VCG cardiac cycle shown for gyration in all three axis components.

## C. Respiratory Variation

Due to the physical proximity of the heart and the lungs, respiration inevitably affects the cardiovascular system. It modulates the morphology, timing and amplitude of cardiac signals, especially mechanical signals such as VCG. This effect was observed as early as the 1920s in some of the first studies of BCG. In these studies, it was noted that respiration phase modulates the amplitude of the acquired heart signal [37]. This view has since been expanded, showing that the respiratory variation of an SCG signal can be explained not only as amplitude modulation but as a combination of amplitude modulation, frequency modulation and baseline wandering. More specifically, the amplitude is dependent on respiratory volume while baseline and frequency modulation are dependent on respiratory phase [56].

Because of this significant respiratory effect, previous works often attempted to reduce the respiratory variation by filtering out the respiration signal altogether [57]. In other works, patients were asked to hold their breath while measurements were taken [58]. While these methods achieve the desired effect of reducing respiratory variation, they also remove potentially useful respiratory information. For example, it has been shown that expiration cycles are affected first in many cardiac abnormalities [59]. Therefore, losing the respiratory information corresponding to these cycles could be detrimental to early detection of such cardiac abnormalities. Additionally, breath holding can be uncomfortable for patients and oftentimes, such as in ambulatory monitoring, it is not a feasible request.

Alternatively, the categorization of SCG heartbeats based on respiratory information such as respiratory volume and respiratory phase has been investigated as a potential method to reduce respiratory variation. This is favourable because it retains useful respiratory information while still reducing respiratory variation. It has been shown that separating heartbeats based on respiratory phase drastically improves analysis and interpretation of the SCG morphology [60]. This method facilitates more accurate estimation of SCG features, and better signal characterization and classification [61], and is the approach taken in this research.

Multiple studies have used ECG or SCG to extract respiratory information such as respiration rate or respiration phase [62, 63], but none have done so on a beat-to-beat basis using VCG. The closest works to this study are [64], in which a machine learning approach was employed on SCG to identify respiratory phase on a beat-to-beat basis, [65], in which the effect

of static respiration volume on VCG signal morphology was investigated, and [58] in which 1-dimensional convolutional neural networks were utilised to classify static lung volume states using VCG. This work builds on the work done in [58].

## D. Machine Learning

The human body is a complex, interconnected system and this causes biomedical signals to be generally prone to noise and interference. As a result, signal characteristics which may provide clinically important information could easily be overlooked by a human [66]. This has led to a significant increase in the use of statistical and computational methods to analyze biomedical signals [67]. The most prevalent of these is machine learning (ML). The benefit of using ML to analyse biomedical signals is that ML has the potential to objectively interpret the signal, in some cases leading to analyses that surpasses that of a human with years of experience [66].

### *1. Supervised learning*

The field of ML consists largely of two main learning paradigms: supervised and unsupervised learning [68]. In supervised learning, a computer program "learns" the correlations between input data and output targets in a training dataset in order to make future predictions about unseen data in a testing dataset. The name "supervised learning" arises from the fact that generally, training is done on a training set where the true output targets are already known. Subsequently, the trained model is used to make predictions on a testing set where the true output targets are unknown. The predictions made by a supervised learning algorithm can be classifications or regressions. Classification attempts to map input data to a discrete set of output targets, or classes, while regression attempts to map input data to a continuous set of output targets. There are many established algorithms for classification such as logistic regression [69], decision trees [70] and support vector machines (SVM) [71]. There are also many established algorithms for regression such as linear regression [72] and support vector regression (SVR) [73].

## *2. Convolutional Neural Networks*

Over the last few decades, a variety of impressive advancements have been brought forth in the field of ML [74]. One of these advancements, and perhaps the most significant, is the evolution of artificial neural networks (ANN). This led to research into increasingly deep neural network architectures with powerful learning capabilities and resulted in the emergence of deep learning (DL) [75, 76] as a sub-field within ML. There are a variety of DL techniques available today, such as recurrent neural networks (RNN) [77] and long short-term memory networks (LSTM) [78]. One of the most popular and widely used DL architectures today is the convolutional neural network (CNN) [79]. CNNs make use of convolving filters that are applied to local features, which ensures a certain degree of shift, scale and distortion invariance [80]. The convolving filters can be applied repeatedly at different layers of the network, giving rise to different features at each layer. This property is important because it allows CNN-based methods to combine feature extraction and prediction tasks into a single body, unlike traditional ML methods. Traditional ML methods generally require certain pre-processing steps be taken to form hand-crafted features which may be computationally expensive and suboptimal. On the other hand, CNN-based methods can extract features directly from raw data and use them to maximise prediction accuracy [81]. This key characteristic improves prediction performance significantly and has made CNN-based methods very attractive for a wide range of complicated applications.

CNN architectures were originally developed for computer vision tasks and therefore were originally proposed only using 2-dimensional (2D) convolving filters. As a result, their application to 1-dimensional (1D) signals was not straightforward. To resolve this, researchers either reshaped 1D signals into a 2D matrix before feeding it to the CNN [82] or simply take a 2D image of the 1D signal and feed that to the CNN [83].

However, utilising 2D convolutions posed a significant computational limitation, thus making the application of 2D CNNs for mobile, low-power or low-memory devices relatively infeasible [81]. In contrast, 1D CNNs offer significantly reduced computational complexity over 2D CNNs. For example, 2D data with $n \times n$ dimensions convolved with a 2D $k \times k$ kernel would have a computational complexity of $\sim O(n^2 k^2)$. Conversely, the corresponding 1D convolution with equivalent dimensions $n$ and $k$ would have a computational complexity of $\sim O(nk)$. This shows that under equivalent conditions (that is, same network configurations,

hyperparameters and kernel size), the computational complexity of a 1D CNN is significantly lower than that of a 2D CNN. Additionally, in order to achieve reasonable generalization capability and high accuracies, 2D CNNs generally require very large datasets. This can pose an issue for many 1D signal applications where available labeled data may not be abundant.

To address these limitations of 2D CNNs, the first adaptive 1D CNNs used on patient-specific ECG signals were proposed [84]. Since then, 1D CNNs have gained significant traction and have even reached state-of-the-art performance in many signal processing domains such as early arrythmia detection in ECG beats [85], damage detection in bearings [86, 87] and high-power motor fault detection [88]. A 1D CNN operates largely the same as a 2D CNN, with the main difference being that 1D convolving filters are used instead of 2D ones. These architectures are especially useful for analysing data along the temporal dimension [89], hence why they were chosen for the task of VCG signal analysis.

## E. 1-Dimensional Convolutional Neural Networks

### *1. Overview*

1D CNNs have many varying architectures but generally follow the framework shown in Figure 3, with an input layer followed by one or more convolution layers, followed by one or more multilayer perceptron (MLP) layers, finally ending with an output or prediction layer. Generally, the convolution layers perform operations on the input and can be viewed as "feature extractor" layers. Consequently, the MLP layers can be viewed as "decision" layers which take the locally extracted features from the convolution layers and use them to make a prediction.

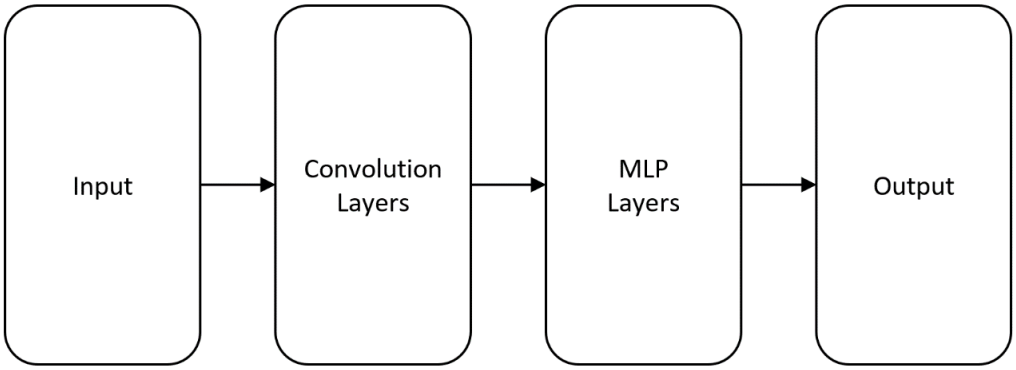


**Figure 3.** A general CNN structure.

### *2. Convolution Layers*

Convolution consists of sliding a kernel over the input signal and computing consecutive dot products with it in a procedure known as shift-compute. For a given 1D input sequence $\boldsymbol{x}$ and a filter $\boldsymbol{k}$ of length $\boldsymbol{2m+1}$, the 1D convolution operation for a given kernel window can be represented by equation (2) to obtain the convolution output $\boldsymbol{h}$.

$$h(t) = (x * k)(t) = \sum_{\tau=-m}^{m} x(t-\tau) \cdot k(\tau) \tag{2}$$

This convolution operation is repeated as the kernel window is shifted across the input. The number of positions shifted after each convolution operation is called the stride. Padding of the input, generally with zeros, is often implemented to handle convolution operations at the edges of the input and ensure consistent dimensionality between input and output.

After the convolution output has been obtained, a bias vector is added, and the resulting value is passed through an activation function (discussed in Section D.5 of Chapter II) to obtain the final output of the layer. Altogether, the output $\boldsymbol{o}$ of a convolution layer for a single kernel (or filter) window is shown in equation (3), where $\boldsymbol{b}$ represents the bias vector and $\boldsymbol{f}(\cdot)$ represents the activation function.

$$o(t) = f\left(b(t) + \sum_{\tau=-m}^{m} x(t-\tau) \cdot k(\tau)\right) \tag{3}$$

This is commonly repeated for multiple filters, giving the final output of the entire convolution layer. These filters, along with the bias term are learned during training to minimize a chosen loss function.

### *3. Multilayer Perceptron Layers*

An MLP layer operates similarly to a 1D convolution layer. However, instead of a filter that is slid across the input, a weight vector is used in the dot product computation with the input. For a given 1D input sequence x and weight vector $\boldsymbol{w}$ of length N, this operation can be represented by equation (4) to obtain output $\boldsymbol{h}$.

$$h(t) = \sum_{t=0}^{N} x(t) \cdot w(t) \tag{4}$$

The output of this operation is also added to a bias vector and passed through an activation function. Altogether, the output $\boldsymbol{o}$ of an MLP layer is shown in equation (5), where $\boldsymbol{b}$ represents the bias vector and $\boldsymbol{f}(\cdot)$ represents the activation function.

$$o(t) = f\left(b(t) + \sum_{t=0}^{N} x(t) \cdot w(t)\right) \tag{5}$$

These layers are stacked and reduced in dimensionality until the output layer is reached, which contains the same number of neurons as classes in the target data. The weight vector and the bias term are learned during training to minimize a chosen loss function.

### *4. Pooling, Dropout and Flatten Layers*

Pooling layers reduce the dimensionality of a given input and highlight its prominent features. Commonly, pooling layers are placed after convolution layers to reduce the dimension of the convolution output. This reduction in dimensionality also helps to reduce overfitting. There are many types of pooling layers, but the most popular type is max pooling. With this type of pooling, a window of a chosen size is slid over the input with a chosen stride and the max value of each window is taken as the output.

Dropout layers employ dropout regularization. The purpose of this technique is to reduce overfitting. Dropout regularization refers to “turning off” one or more neurons in the network at random, such that they provide zero input to the subsequent layer. This prevents an over-reliance on a few of the neurons and forces the network to learn to use as many neurons as possible, thus improving generalization. The number of neurons that are turned off is determined by a probabilistic measure called the dropout rate. For example, if the dropout rate is 0.5, a number between 0 and 1 will be randomly assigned to each neuron and those with a value lower than the dropout rate will be turned off.

The output of convolution layers may have a depth greater than one. Therefore, in order for this output to be passed to an MLP layer, it must first be flattened. Flatten layers simply

concatenate the output from a convolution layer to form a flat structure which can then be passed to an MLP layer.

### *5. Activation Functions*

Activation functions are used largely to introduce non-linearity into a neural network and allow it to model non-linear correlations. There exist linear activation functions, but these are not as widely used as non-linear ones [90]. If linear activation functions were used (or if activation functions were omitted altogether), the output of the network would effectively be a linear combination of the network layers. Therefore, the network would only be able to adapt to linear changes of the input. However, real world data possess non-linear characteristics which also need to be considered. Therefore, non-linear activation functions are generally preferred. In this research, the activation function was taken as a hyperparameter to be tuned during experimentation. This section describes some common activation functions.

One of the earliest popular activation functions was the sigmoid function. This activation function transforms the input values to the range of 0 to 1 and is given by equation (6).

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

Another popular activation function is the hyperbolic tangent function. This function is similar to the sigmoid function, with the main difference being that it is symmetric around the origin, transforming input values to the range of -1 to 1 instead of 0 to 1 like sigmoids. This activation function is given by equation (7).

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{7}$$

However, the current most popular activation function is the rectified linear unit (ReLU). This activation function typically converges faster than the sigmoid activation function and has even been shown to be an average of cascaded sigmoids [91]. ReLU activation is given by equation (8).

$$\mathrm{ReLU(x)} = \begin{cases} \mathrm{x}, & \text{if } \mathrm{x} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Additionally, activation functions for the output layer depend on the type of output. For a regression analysis output, any of the aforementioned activation functions could be used. However, for a classification analysis output, a specific type of output activation function is required. This activation function is called the SoftMax function [90]. This function creates a probability distribution over $n$ classes, assigning a weighted probability to each class based on the output in that class's position. This function ensures that all the output values will sum to 1, thus constituting a valid probability distribution. The SoftMax activation function for a given class $i$ out of $n$ classes is given by equation (9).

$$SoftMax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{9}$$

## *6. Training*

In order to train a 1D CNN, the trainable parameters within the network (that is, the weights, filters and biases) must be learned based on the training data. This is done by defining a loss function and minimizing it in a process called back-propagation to update these parameters. Loss functions are generally convex functions, which means that updating parameters in a direction opposite to the gradient of the loss function with respect to the parameters will result in them reaching a global minimum of the loss function. These gradients are backpropagated through the network in an iterative optimization process called gradient descent. For gradient descent, a constant learning rate which is defined before training is used to control how quickly the parameters are updated based on the gradients. However, there are other optimizers such as Adam [92], RMSprop [93], or momentum [94] which alter their learning rate over time in a process called adaptive learning. While many of these adaptive learning optimizers have been shown to converge faster than regular gradient descent, in this research, the type of optimizer used in the model was taken as a hyperparameter to be tuned during experimentation.

## F. Thesis Objectives

The objective of this thesis was to determine whether 1D CNNs are a viable tool to classify VCG cardiac cycles based on the respiratory volume and phase of the subject at the time they were recorded. To achieve this, three classification objectives were defined: classifying static respiratory volume state, classifying dynamic respiratory volume state, and classifying dynamic respiratory phase. The collected VCG data were split into two classes for each of the classification objectives, based on the concurrently recorded respiratory flow signal. For each classification objective, 1D CNN was employed to predict the class of each cardiac cycle.

Subsequently, the secondary objective of this thesis was to optimize the performance of each 1D CNN for each classification objective through experimentation. This was achieved through hyperparameter tuning of the models. Various values for each of the relevant hyperparameters were investigated until a final optimal architecture for each classification objective was found.

The development of the data acquisition system, data collection methods, data processing, feature and target construction methods and all analysis and experimentation steps are detailed in Chapter II. The results are presented in Chapter III and discussed in Chapter IV. The thesis is concluded in Chapter V and Chapter VI presents an appendix.

# II. METHODS

## A. Data Collection

Experimental data were collected at McGill University with approval from the McGill Review Ethics Board. Data were collected using a custom-built system using commercially available and affordable components that could be assembled with minimal complexity. This facilitated a set up that could easily be replicated in other labs, thus demonstrating the broad applicability of VCG.

### *1. Sensors*

Cardiac activity was recorded by a 6 axis IMU (MPU 9250, InvenSense) attached to the xiphoid process of the sternum with a single piece of double-sided tape. The MPU 9250 was selected for its small form factor. Moreover, it facilitates digital sampling with a sampling rate up to 4000 Hz and with a low root mean square (RMS) noise of $0.078\ ms^{-2}$ [95]. The range of the accelerometer was set to $\pm\ 2\ g$ while the range of the gyroscope was set to $\pm\ 250\ deg/sec$. The positive X, Y and Z-axes of the accelerometer were oriented downward, right and outward respectively. Consequently, the gyroscope coordinates followed the right-hand rule for rotation about these axes.

A Biopac digital acquisition system (MP160, Biopac) was used to obtain reference ECG (BN-RSPEC, Biopac) and spirometer (TSD137H, Biopac) signals. Both systems have been validated for clinical use. The ECG electrodes were attached to the skin in an Einthoven triangle [96] on the torso. The spirometer with a disposable mouthpiece filter was placed in the subject's mouth. Before testing each subject, the spirometer was calibrated to a 3-litre reference syringe (AFT27, Biopac). The described placement and orientation of the IMU, ECG and spirometer sensors is shown in Figure 4(a).

### *2. System Connectivity*

The IMU data were polled over inter-integrated circuit (I2C) by a Raspberry Pi (Pi-Zero W, Raspberry Pi) at a sampling rate of approximately 600 Hz. A Raspberri Pi was chosen

because it facilitated wireless communications, a sufficiently powerful processor and quick iterations of prototyping. The Raspberry Pi was controlled by a custom-built webserver user interface designed in Python [97] using Flask [98]. The polled IMU data were saved to a text file on the Raspberry Pi and then transferred wirelessly to a laptop computer via file transfer protocol (FTP). The Biopac data were transmitted serially to a second computer and recorded and processed with the AcqKnowledge 5.0 (ACK100W, Biopac) software provided with the Biopac system. This software was used for standard smoothing and filtering of the acquired Biopac signals to mitigate sensor noise. Moreover, ECG annotation routines within the software were used to identify R-peaks.

Additionally, an externally wired clock signal with varying pulse widths was sent from the Biopac system to the Raspberry Pi to create a global timing reference between the two systems and facilitate easier synchronisation in processing. Data were combined, synchronized and processed in Matlab. A schematic of the overall system and its connections is shown in Figure 4(b).

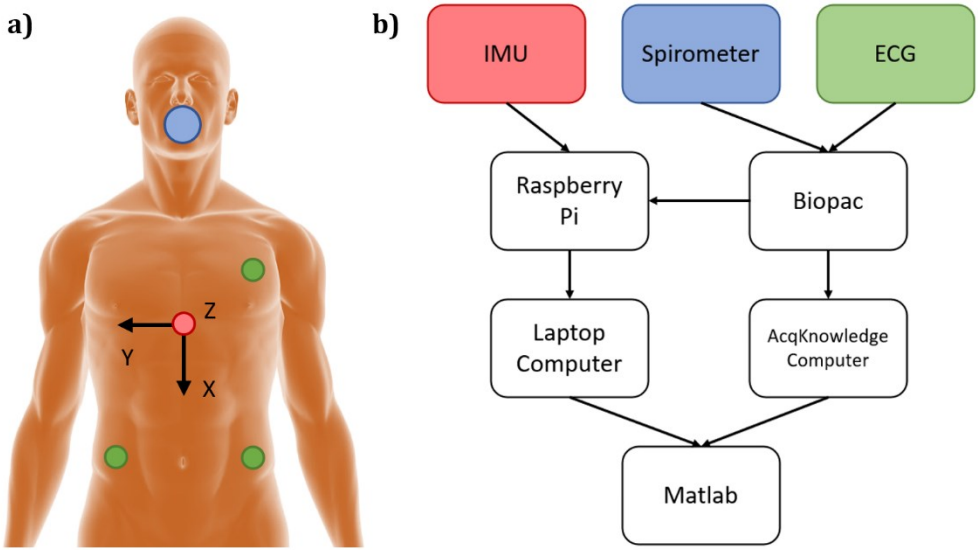


**Figure 4.** (a) Placement of the inertial measurement unit (IMU) on the xiphoid process of the sternum (shown in red) with its orientation represented by the Cartesian reference axis (shown in black), placement of the spirometer (shown in blue) and placement of the electrocardiography (ECG) electrodes attached to the torso (shown in green). (b) Diagram of data flow.

### *3. Human Trials*

Subjects were first asked to keep a tight seal around the spirometer mouthpiece filter and breathe normally through their mouth. No other instructions were given to the subject to regulate the rate, intensity or depth of their breathing. Spirometer, ECG and VCG signals were recorded for 3 minutes while the subject was at rest. This constituted one "rest" test. Each subject performed two rest tests. Next, subjects were asked to hold their breath as long as possible at both high lung volume (HLV) and low lung volume (LLV), with a maximum of 2 minutes for HLV and 1 minute for LLV. The HLV holds involved inhaling as much as possible before holding, while the LLV holds involved exhaling as much as possible before holding. These holds were each repeated twice more with rests in between, giving a total of three HLV holds and three LLV holds per subject. All tests were performed with the subject in the supine position. The study was conducted on 50 healthy participants for a total of 18,330 cardiac cycles at rest and 15,619 cardiac cycles of breath holds. The average metrics of the study population can be seen in Table 1. All subjects had no known prior or existing respiratory or cardiovascular conditions, and they all signed a consent form at the beginning of the study.

Table 1: Study Population

| **Description** | **Value** |
|---|---|
| Participants | 50 |
| Percent Male | 56% |
| Age | $24.4 \pm 4.45$ years |
| Weight | $69.1 \pm 13.0$ kg |
| Height | $172.6 \pm 10.6$ cm |

## B. Data Processing

After acquisition, all data were processed using Biopac's AcqKnowledge 5.0 software, Matlab (R2020A) and Python. In the AcqKnowledge 5.0 software, a modified Pan Tomkins real-time QRS analysis algorithm [99] supplied by the software was used to detect R-peaks in the

ECG signal. However, this algorithm incorrectly identified some of the R-peaks, which was remedied in Matlab by locating the incorrectly identified R-peaks and reassigning them to the correct peak.

In Matlab, the varying pulse widths in the reference clock signal were used to identify the start and end points of all signals from the IMU and Biopac systems and match them in time. Due to the inherent time discrepancies caused by manually starting the IMU and Biopac recordings for each test, the signals often had portions either at the start or the end which did not overlap with the other signals for a given test. Therefore, each signal was carefully trimmed to ensure that only the portions which overlapped with the other signals were retained. Subsequently, the respiratory flow signal from the spirometer was smoothed and integrated to measure respiratory volume. This numerical integration introduced drift to the signal, which was removed by fitting a $2^{nd}$ order polynomial to the recording and subtracting it. Additionally, due to the large random spikes in the VCG signal from cardiac vibrations, spikes that were larger than 5 times the median of the signal were treated as outliers and removed. The time-matched, trimmed and processed signals were then exported to Python for further processing.

In Python, the VCG data from the IMU was interpolated to 200 Hz for faster post-processing and to match the sampling rate needed for SCG analyses [40]. A $5^{th}$ order Butterworth filter was used to remove high-frequency noise.

As mentioned, CNNs can extract features directly from raw data and use them to maximise prediction accuracy. They also tend to perform better when given more data. Therefore, all 6 axes of the VCG signal were used in feature construction.

## C. Feature and Target Construction

Feature and target construction involved separating VCG signals into cardiac cycles and then using either the location of each cardiac cycle on the respiratory volume signal or the static breath hold test each cardiac cycle came from to create respiratory targets.

### *1. VCG Cardiac Cycles*

The VCG data were separated into cardiac cycles using the concurrently recorded ECG signal. The beginning of each cardiac cycle was set as 0.02 seconds prior to the timestamp of the R-peak in the ECG signal. This was done to approximately account for the onset of the P wave and consequently, the vibrations corresponding to the given cardiac cycle. Each cardiac cycle was then padded to a uniform length of 500 samples per cardiac cycle with the average value of the cycle. Previously, the cycles were interpolated to 500 samples instead of padded. This approach was abandoned because each cycle was then stretched differently in time, leading to the loss of potentially useful timing information. During experimentation it was found that padding the cardiac cycles instead of interpolating them led to a greater prediction accuracy.

The uniform-length cardiac cycle vectors were then concatenated to form a preliminary $n \times m$ feature matrix for each axis component, where $n$ was the number of cardiac cycles in the dataset and $m$ was the number of elements per cardiac cycle (500 in this case). The preliminary feature matrix is shown in equation (1), where $x_n[m]$ represents the $m^{th}$ element of the $n^{th}$ cardiac cycle.

$$\begin{bmatrix} x_1[1] & x_1[2] & \cdots & x_1[m] \\ \vdots & \vdots & \ddots & \vdots \\ x_n[1] & x_n[2] & \cdots & x_n[m] \end{bmatrix} \tag{1}$$

This process was repeated for all six axis components, and the resulting feature arrays were concatenated along a third axis to form the final $n \times m \times 6$ feature matrix used for training.

### *2. Static Respiratory Volume Targets*

For predicting static respiratory volume, the breath hold test data were used. The target creation process was trivial, and the targets were binary; with a 1 attributed to cardiac cycles taken from an HLV test and a 0 attributed to cardiac cycles taken from an LLV test.

### *3. Dynamic Respiratory Volume Targets*

For predicting dynamic respiratory volume, the rest test data were used. The location of the start of each cardiac cycle on the respiratory volume signal was found and the volume at each

location was examined. Cardiac cycles with a volume above the mean of the respiratory volume signal were taken as HLV cycles and attributed a 1, while cardiac cycles below the mean of the respiratory volume signal were taken as LLV cycles and attributed a 0. The HLV and LLV cycles are shown in Figure 5, overlayed onto the respiratory flow signal.

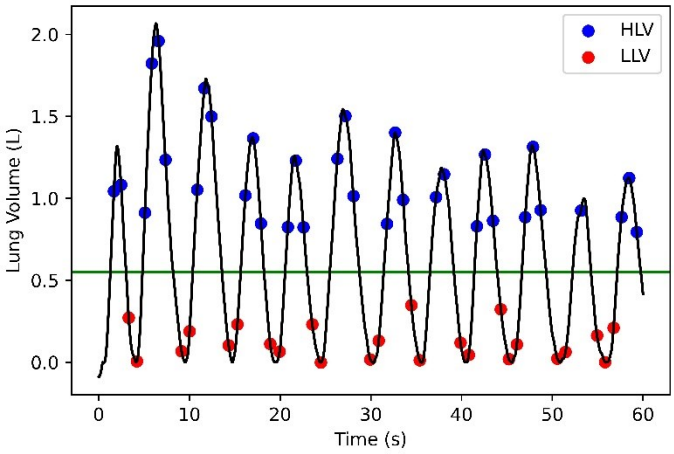


**Figure 5.** HLV and LLV cardiac cycles overlayed on respiratory flow signal.

### *4. Dynamic Respiratory Phase Targets*

For predicting dynamic respiratory phase, the rest test data were also used. The location of each cardiac cycle on the respiratory volume signal was found and the slope of the respiratory volume signal at each location was examined. Cardiac cycles on a positive slope were taken as inspiration cycles and attributed a 1, while cardiac cycles on a negative slope were taken as expiration cycles and attributed a 0. The inspiration and expiration cycles are shown in Figure 6, overlayed onto the respiratory flow signal.

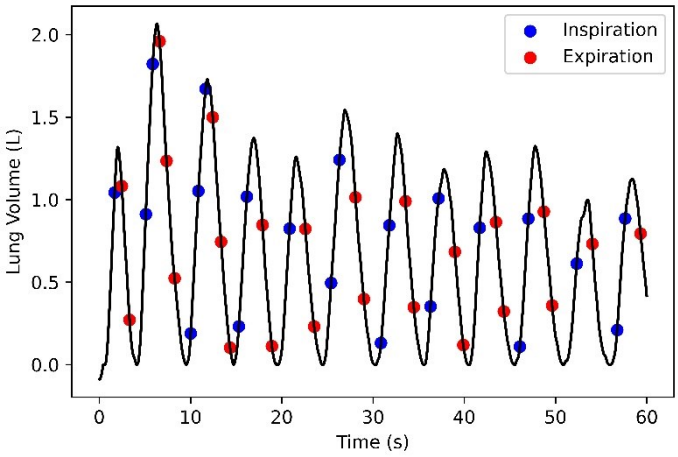


**Figure 6.** Inspiration and expiration cardiac cycles overlayed on respiratory flow signal.

## D. Experimentation and Evaluation

### *1. Hyperparameter Tuning*

There are many factors which can affect the performance of a 1D CNN. As mentioned, there are parameters such as weights, filter weights and biases which are learned during training of the network. However, there are also various parameters that affect performance which are not learned during training. These parameters are called hyperparameters. Generally, experimentation to improve the performance of a 1D CNN involves tuning these hyperparameters to achieve higher performance. In this research, a base architecture from [58] was selected due to its proven success in respiratory classification from VCG signals. This base architecture is shown in Figure 7.

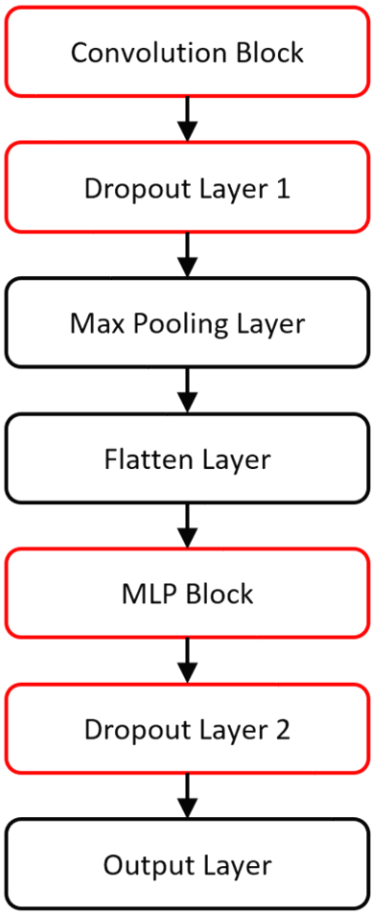


**Figure 7.** Base architecture upon which hyperparameter tuning was performed. Red blocks represent blocks that were changed during hyperparameter tuning, black blocks represent blocks that were unchanged.

Subsequently, various hyperparameters on this base architecture were experimented on until a final optimal architecture for each of the chosen classification objectives was found. First, each hyperparameter was investigated in isolation, leaving the other hyperparameters unchanged from the base architecture. Subsequently, the highest performing values from each hyperparameter search were combined to form the optimal architecture for each classification objective. The optimal architecture was then evaluated using the techniques detailed in Section E.2 of Chapter II. This process was repeated using the data for each of the classification objectives. The hyperparameters investigated are discussed in this section. The values of these hyperparameters for the base architecture are shown in Table 2.

**Number of convolution layers** – The number of convolution layers included in the convolution block.

**Number of MLP layers** – The number of MLP layers included in the MLP block.

**Number of convolution filters** – The number of convolution filters trained for each convolution layer.

**Batch size** – The number of training samples propagated through the network in order to update the trainable parameters.

**Number of epochs** – The number of times training data are propagated through the network and errors are backpropagated to update trainable parameters. A single epoch refers to a single iteration of propagation followed by a single iteration of backpropagation.

**Activation functions** – The activation functions used for convolution and MLP layers. The activation function at the output remains unchanged because the objective is always a classification task.

**Filter/kernel size** – The width of the convolution filters used in the convolution layers. Due to the nature of the convolution operation, this value must always be an odd integer.

**Optimizer** – The optimization algorithm used to iteratively update the trainable parameters of the network based on the loss function.

**Loss function** – The loss function used by the optimization algorithm to iteratively update the trainable parameters of the network.

**Dropout rates** – The rate at which neurons are “dropped” in a given dropout layer. In the base architecture, there is a dropout layer after the convolution block and another one after the MLP block. These are the two dropout rates investigated in this research.

**Number of MLP layer neurons** – The number of neurons in each of the MLP layers of the network.

Table 2: Base architecture hyperparameter values

| **Hyperparameter** | **Value** |
|---|---|
| Number of convolution layers | 2 |
| Number of MLP layers | 1 |
| Number of convolution filters | 512 |
| Batch size | 50 |
| Number of epochs | 50 |
| Activation function – Conv | ReLU |
| Activation function – MLP | ReLU |
| Filter/kernel size | 3 |
| Optimizer | Adam |
| Loss function | Sparse categorical cross-entropy |
| Dropout rate – Conv | 0.5 |
| Dropout rate – MLP | 0.5 |
| Number of MLP neurons | 100 |

## *2. Evaluation Techniques*

When evaluating the performance of a binary classification model, predictions are divided into four possible outcomes. Positive targets correctly predicted as positive are labelled true positives (TP), negative targets correctly predicted as negative are labelled as true negatives (TN), positive targets incorrectly predicted as negative are labelled false negatives (FN) and negative targets incorrectly predicted as positive are labelled false positives (FP). Using these outcomes, evaluation metrics such as accuracy, precision, recall and F1 score can be defined. These metrics are given by equations (10), (11), (12) and (13) respectively. Accuracy is the fraction of correct predictions overall, precision is the fraction of all positive predictions that were actually positive targets and recall is the fraction of all positive targets that were correctly

predicted as positive. Additionally, F1 score combines accuracy, precision and recall into one metric.

$$\text{Accuracy} = \frac{\sum TP + \sum TN}{\sum \text{Total Population}} \tag{10}$$

$$\text{Precision} = \frac{\sum TP}{\sum TP + \sum FP} \tag{11}$$

$$\text{Recall} = \frac{\sum TP}{\sum TP + \sum FN} \tag{12}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \sum TP}{2 \sum TP + \sum FP + \sum FN} \tag{13}$$

To evaluate the performance of a given CNN architecture and its hyperparameters, K-fold cross validation was used [100]. This method of evaluation splits the dataset into $k$ folds of equal size, then trains on $k - 1$ folds and tests on the remaining fold. This process is repeated $k$ times, each time changing the testing fold such that each of the $k$ folds is used as a testing fold throughout the process. The average evaluation metrics across the $k$ iterations is used as the measure of performance. This method of evaluation is beneficial because validating performance on multiple folds of data helps to detect and thus mitigate overfitting.

# III. RESULTS

This chapter describes the results obtained from the analyses described above. The value of each hyperparameter was varied using a predetermined set of options and 10-fold cross validation was performed to obtain an average F1 score for each value. Subsequently, the value with the highest F1 score was included in the final optimal architecture for the given classification objective.

## A. Static Respiratory Volume

For the static respiratory volume classification objective, the best and worst performing hyperparameter values are shown in Table 3. A more detailed table showing the results of all values tested for all hyperparameters is shown in Appendix A.

Table 3: Average F1 score for the best and worst performing values of each hyperparameter tuning experiment from the static respiratory volume classification objective.

| **Hyperparameter** | **Values Tested** | **Best** | | **Worst** | |
|---|---|---|---|---|---|
| | | **Value** | **Avg F1 Score** | **Value** | **Avg F1 Score** |
| Number of convolution layers | 1, 2, 3, 4, 5 | 1 | 0.9947 | 3 | 0.9937 |
| Number of MLP layers | 1, 2, 3, 4, 5 | 1 | 0.9949 | 5 | 0.9939 |
| Number of convolution filters | 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 | 16 | 0.9950 | 2 | 0.9855 |
| Batch size | 32, 64, 128, 256 | 128 | 0.9950 | 64 | 0.9947 |
| Number of epochs | 50, 100, 150, 200, 250, 300 | 300 | 0.9948 | 50 | 0.9938 |
| Activation function (Conv, MLP) | ReLU, Sigmoid, Tanh | (ReLU, ReLU) | 0.9946 | (Tanh, Sigmoid) | 0.9185 |
| Filter/kernel size | 3, 5, 7, 9, 11, 13, 15, 17, 19 | 7 | 0.9960 | 3 | 0.9944 |
| Optimizer | SGD, RMSprop, Adam, Adamax, Nadam, Adadelta | SGD | 0.9956 | Adadelta | 0.9746 |
| Loss function | Sparse Categorical Cross-entropy, Poisson, KL Divergence | Sparse Categorical Cross-entropy | 0.9945 | KL Divergence | 0.1606 |

| | | | | | |
|---|---|---|---|---|---|
| Dropout rate (Conv, MLP) | 0.1, 0.3, 0.5, 0.7, 0.9 | (0.5, 0.1) | 0.9950 | (0.9, 0.9) | 0.9781 |
| Number of MLP neurons | 10, 50, 100, 250, 500, 750, 1000 | 500 | 0.9954 | 10 | 0.9903 |

These findings correspond with previous work [58] from which the base architecture for these experiments was obtained. However, it was found that a reduction in the number of convolution layers from 2 to 1 resulted in a 0.02% increase in F1 score from 0.9945 to 0.9947. It was also found that a reduction in the number of convolution filters from 512 to 16 resulted in a 0.05% increase in F1 score from 0.9945 to 0.9950. Additionally, it was observed that a larger kernel size, more epochs, more MLP neurons, a smaller MLP dropout rate and a change in optimizer all lead to an increase in performance. In fact, all of the best performing hyperparameter values in these experiments produced results that either matched or outperformed the F1 score from that previous work. Moreover, the optimal architecture for this classification objective achieved an F1 score of 0.9972. This was an increase of 0.12% over the best F1 score from the individual hyperparameter searches and 0.27% over the base architecture's performance on this classification objective. Table 4 shows the performance of the optimal architecture in comparison with the performance of the base architecture.

Table 4: Performance of optimal architecture and base architecture for the static respiratory volume classification objective.

| **Model** | **F1 Score** |
|---|---|
| Optimal architecture | 0.9972 |
| Base architecture | 0.9945 |

## B. Dynamic Respiratory Volume Targets

For the dynamic respiratory volume classification objective, the best and worst performing hyperparameter values are shown in Table 5. A more detailed table showing the results of all values tested for all hyperparameters is shown in Appendix B.

Table 5: Average F1 score for the best and worst performing values of each hyperparameter tuning experiment from the dynamic respiratory volume classification objective.

| Hyperparameter | Values Tested | Best Value | | Worst Value | |
|---|---|---|---|---|---|
| | | Value | Avg F1 Score | Value | Avg F1 Score |
| Number of convolution layers | 1, 2, 3, 4, 5 | 1 | 0.8052 | 5 | 0.7924 |
| Number of MLP layers | 1, 2, 3, 4, 5 | 1 | 0.8034 | 2 | 0.7863 |
| Number of convolution filters | 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 | 16 | 0.8019 | 2 | 0.7095 |
| Batch size | 32, 64, 128, 256 | 64 | 0.8159 | 256 | 0.7958 |
| Number of epochs | 50, 100, 150, 200, 250, 300 | 300 | 0.8071 | 50 | 0.7805 |
| Activation function (Conv, MLP) | ReLU, Sigmoid, Tanh | (ReLU, ReLU) | 0.8098 | (Sigmoid, Tanh) | 0.3492 |
| Filter/kernel size | 3, 5, 7, 9, 11, 13, 15, 17, 19 | 15 | 0.8144 | 5 | 0.8028 |
| Optimizer | SGD, RMSprop, Adam, Adamax, Nadam, Adadelta | SGD | 0.8098 | Adadelta | 0.7141 |
| Loss function | Sparse Categorical Cross-entropy, Poisson, KL Divergence | Sparse Categorical Cross-entropy | 0.8048 | KL Divergence | 0.0619 |
| Dropout rate (Conv, MLP) | 0.1, 0.3, 0.5, 0.7, 0.9 | (0.5, 0.5) | 0.8131 | (0.9, 0.9) | 0.5975 |
| Number of MLP neurons | 10, 50, 100, 250, 500, 750, 1000 | 500 | 0.8123 | 10 | 0.7679 |

In these experiments, it was again found that using just one convolution layer produced the best results. Additionally, it was found that fewer convolution filters, a smaller batch size, more epochs, a larger kernel size and more MLP neurons all improved performance. Moreover, the optimal architecture for this classification objective achieved an F1 score of 0.8324. This was an increase of 2.02% over the best F1 score from the individual hyperparameter searches and 4.01% over the base architecture's performance on this classification objective.

Table 6 shows the performance of the optimal architecture in comparison with the performance of the base architecture.

Table 6: Performance of optimal architecture and base architecture for the dynamic respiratory volume classification objective.

| Model | F1 Score |
|---|---|
| Optimal architecture | 0.8324 |
| Base architecture | 0.8003 |

## C. Dynamic Respiratory Phase Targets

For the dynamic respiratory phase classification objective, the best and worst performing hyperparameter values are shown in Table 7. A more detailed table showing the results of all values tested for all hyperparameters is shown in Appendix C.

Table 7: Average F1 score for the best and worst performing values of each hyperparameter tuning experiment from the dynamic respiratory phase classification objective.

| Hyperparameter | Values Tested | Best Value | | Worst Value | |
|---|---|---|---|---|---|
| | | Value | Avg F1 Score | Value | Avg F1 Score |
| Number of convolution layers | 1, 2, 3, 4, 5 | 2 | 0.8410 | 5 | 0.8370 |
| Number of MLP layers | 1, 2, 3, 4, 5 | 1 | 0.8385 | 2 | 0.8279 |
| Number of convolution filters | 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 | 16 | 0.8429 | 2 | 0.7857 |
| Batch size | 32, 64, 128, 256 | 32 | 0.8409 | 64 | 0.8345 |
| Number of epochs | 50, 100, 150, 200, 250, 300 | 300 | 0.8410 | 50 | 0.8324 |
| Activation function (Conv, MLP) | ReLU, Sigmoid, Tanh | (ReLU, Sigmoid) | 0.8459 | (Sigmoid, Tanh) | 0.7723 |
| Filter/kernel size | 3, 5, 7, 9, 11, 13, 15, 17, 19 | 7 | 0.8430 | 17 | 0.8319 |
| Optimizer | SGD, RMSprop, Adam, Adamax, Nadam, Adadelta | Nadam | 0.8476 | Adadelta | 0.7628 |
| Loss function | Sparse Categorical Cross-entropy, Poisson, KL Divergence | Sparse Categorical Cross-entropy | 0.8435 | KL Divergence | 0.0685 |
| Dropout rate (Conv, MLP) | 0.1, 0.3, 0.5, 0.7, 0.9 | (0.5, 0.3) | 0.8415 | (0.9, 0.9) | 0.6950 |
| Number of MLP | 10, 50, 100, 250, 500, | 750 | 0.8471 | 10 | 0.8119 |

| | | | | | |
|---|---|---|---|---|---|
| neurons | 750, 1000 | | | | |

In these experiments, it was found that fewer convolution filters, a smaller batch size, more epochs, a change in MLP activation function, a larger kernel size, a change in optimizer and more MLP neurons all improved performance over the base architecture. Moreover, the optimal architecture for this classification objective achieved an F1 score of 0.8617. This was an increase of 1.66% over the best F1 score from the individual hyperparameter searches and 4.66% over the base architecture's performance on this classification objective. Table 8 shows the performance of the optimal architecture in comparison with the performance of the base architecture.

Table 8: Performance of optimal architecture and base architecture for the dynamic respiratory phase classification objective.

| **Model** | **F1 Score** |
|---|---|
| Optimal architecture | 0.8617 |
| Base architecture | 0.8233 |

# IV. DISCUSSION

The optimal architectures developed during experimentation provided reliable classification performance for all three of the classification objectives outlined, showing that 1D CNNs are a viable tool to classify VCG cardiac cycles based on the respiratory volume and phase of the subject at the time they were recorded. Due to the high F1 scores achieved, the results of this research validate and expand upon the conclusions reached by [65], in which it was shown that differences in lung volume state result in a quantifiable distinction between VCG waveforms.

One important finding of this research is that networks with fewer convolution layers tended to outperform networks with more convolution layers. This is highlighted by the fact that none of the three classification objectives produced an optimal architecture with more than two convolution layers, despite up to five convolution layers being tested for each one. As discussed in Section D.2 of Chapter I, convolution layers can be viewed as “feature extractor” layers which apply convolving filters to local features. That is, each filter of a convolution layer receives inputs from a set of filters located in a small neighbourhood in the previous layer. These filters can be applied repeatedly at different layers of the network to extract different features at each layer, each one building upon features extracted at previous layers. For example, in the case of image classification with CNNs, lower-level convolution layers have been shown to extract elementary visual features such as edges, corners and endpoints of an image. Subsequent layers build upon and combine these lower-level elementary features to extract more complex features, such as detecting certain objects based on the specific combination of edges, corners and endpoints found in those objects [101]. In other words, more convolution layers in a model suggests that features with increasing levels of complexity are being extracted before the decision-making process. However, the fact that models with fewer convolution layers outperformed models with more convolution layers suggests that the less complex feature representations extracted in the lower-level convolution layers were adequate to make accurate predictions. In fact, this finding suggests that further computation after these lower-level layers led to wasted computational power and degradation of model performance. In the context of CNNs, this degradation of model performance and wastage of computational resources is commonly referred to as overthinking [102]. One potential cause of overthinking in this research

is that higher-level, more complex feature representations are overlapping between classes more frequently than lower-level, less complex feature representations. That is, more of the higher-level features exist in both classes than the lower-level features, thus giving the lower-level features a greater ability to distinguish between classes. Another potential cause of overthinking in this research is that higher-level features are focusing on finer details in the signal that are largely irrelevant to the prediction being made, thus giving the lower-level features (that lack such irrelevant focuses) a greater ability to distinguish between classes. The true cause of overthinking in this research is unknown and could be a combination of the two potential causes discussed. This is a potential avenue for further research into lung volume-based VCG analysis with 1D CNNs.

Another important finding of this research is that decreasing the number of convolution filters in the model lead to greater performance. This is highlighted by the fact that the optimal architecture produced for all three classification objectives used 16 convolution filters, a decrease from 512 in the base architecture. As discussed in Section D.2 of Chapter I, multiple convolution filters can be added to each convolution layer of the network. The weights of each filter are initialized randomly and learned during training, generally leading to a collection of filters wherein each filter learns to extract a specific feature in the data. For example, again in the case of image classification with CNNs, each feature (such as edges, corners or endpoints) is extracted by a corresponding convolution filter. Therefore, more convolution filters in a given convolution layer suggests that at that level of abstraction, more features are being extracted. As a result, the fact that decreasing the number of convolution filters lead to greater performance suggests that fewer individual features were required to make an accurate prediction for all three classification objectives. In fact, this finding shows that extracting more features per layer after a certain point degraded model performance. This point was found to be about 16 filters, up to which performance increased but after which performance degraded. To determine the true cause of the underperformance of models with more convolution filters, further research is required. However, one potential cause is that there are approximately 16 important features in the data that correlate strongly to the respiratory variation in the signal. Therefore, models with more filters which attempt to extract more features will likely extract features which are not relevant to respiratory variation. As a result, they introduce noise into the decision-making process, leading to degradation of model performance.

The implications of these findings are favourable. Removing convolution layers from a model decreases the number of parameters which must be learned during training. For example, the base architecture modified to have 5 convolution layers has 15,701,806 trainable parameters, while the base architecture modified to have 1 convolution layer has 12,758,830 trainable parameters, an 18.74% decrease. This decrease is favourable because fewer trainable parameters can increase training by requiring fewer iterations and fewer computations per iteration [103]. In addition, fewer convolution filters in a model decreases the number of filter weights that must be learned during training and thus also reduces the overall number of trainable parameters. For example, the base architecture unchanged with 512 convolution filters for each convolution layer has 13,494,574 trainable parameters, while the base architecture modified to use 16 convolution filters for each convolution layer has 398,190 trainable parameters, a decrease of 97.05%. This decrease is even more significant than that of reducing the number of convolution layers. These reductions in training time and computational complexity could improve feasibility of the model being used for mobile or low-power applications.

Additionally, several limitations of the conducted study have been recognized. To begin with, the study population was only 50 subjects. In order to properly validate the findings of this study, it would be beneficial that a much larger study population be used. Secondly, all data were recorded with the subject laying supine in a controlled environment with minimal external noise affecting the recordings. If VCG is to be developed into a robust non-invasive health monitoring tool, it must also be investigated in scenarios more aptly emulating everyday life. Thirdly, other than setting accelerometer and gyroscope ranges, no further calibration was performed on the IMU systems before each experiment. This is an area for future improvement, as calibrating on a per-experiment basis would likely yield more reliable results. Furthermore, due to time constraints, no specific investigations were carried out into the overall fault-tolerance or production deployment potential of the system. In order for the respiratory variation reducing CNNs in this work to be incorporated into a successful productionized wearable VCG solution, their reliability under condition of various faults (such as sensor faults or computing node faults) must be investigated and maximised, and their application to a fully online production system must be engineered. Finally, the inherent nature of 1D CNNs and most ML techniques is that they give little insight into the correlations being made within the model. That is, while proving to be very accurate in their predictions, little can be gleaned about how exactly those predictions

are being made. Therefore, an excellent future extension of this study would be to recreate the work done with a custom-designed 1D CNN that provides output at each inner layer, thus facilitating deeper analysis of the correlations being made within the model.

# V. CONCLUSION

The objective of this thesis was to determine whether 1D CNNs are a viable tool to classify VCG cardiac cycles based on the respiratory volume and phase of the subject at the time they were recorded. Three classification objectives were defined: classifying static respiratory volume state, classifying dynamic respiratory volume state and classifying dynamic respiratory phase. It was found that a 1D CNN was able to classify VCG cardiac cycles for all three classification objectives with a high degree of accuracy. Moreover, it was found that this accuracy could be increased by reducing the complexity of the models in all three cases. These findings have positive implications for the application of 1D CNNs to the task of VCG analysis, increasing the viability of deploying such models in low-power, low-memory or mobile applications. Further studies must be conducted with larger study populations and longer breathing windows to validate these findings. Additionally, future work could involve repeating these analyses with custom-built 1D CNNs that can facilitate deeper analysis into the correlations being made within the model. This would enable a deeper understanding of which aspects of the VCG signal were most important to the classifications that were made.

# VI. APPENDIX

## A. Individual Hyperparameter Search Results – Static Respiratory Volume

1. Number of Convolution Layers

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | Average | Standard Deviation |
| 1 | **0.9947** | 0.0018 |
| 2 | 0.9943 | 0.0013 |
| 3 | 0.9937 | 0.0013 |
| 4 | 0.9945 | 0.0012 |
| 5 | 0.9942 | 0.0017 |

2. Number of MLP Layers

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | Average | Standard Deviation |
| 1 | **0.9949** | 0.0011 |
| 2 | 0.9944 | 0.0017 |
| 3 | 0.9940 | 0.0016 |
| 4 | 0.9948 | 0.0015 |
| 5 | 0.9939 | 0.0019 |

3. Number of Convolution Filters

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | Average | Standard Deviation |
| 2 | 0.9855 | 0.0021 |
| 4 | 0.9989 | 0.0014 |
| 8 | 0.9937 | 0.0020 |
| 16 | **0.9950** | 0.0007 |
| 32 | 0.9946 | 0.0016 |

| | | |
|---|---|---|
| 64 | 0.9948 | 0.0022 |
| 128 | 0.9944 | 0.0016 |
| 256 | 0.9948 | 0.0010 |
| 512 | 0.9943 | 0.0017 |
| 1024 | 0.9944 | 0.0017 |

4. Batch Size

| **Hyperparameter Value** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 32 | 0.9949 | 0.0017 |
| 64 | 0.9947 | 0.0016 |
| 128 | **0.9950** | 0.0012 |
| 256 | 0.9947 | 0.0015 |

5. Number of Epochs

| **Hyperparameter Value** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 50 | 0.9938 | 0.0010 |
| 100 | 0.9944 | 0.0016 |
| 150 | 0.9943 | 0.0005 |
| 200 | 0.9944 | 0.0021 |
| 250 | 0.9947 | 0.0017 |
| 300 | **0.9948** | 0.0015 |

6. Activation Function

| **Hyperparameter Value (Conv, MLP)** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| (ReLU, ReLU) | **0.9946** | 0.0014 |
| (ReLU, Sigmoid) | 0.9939 | 0.0019 |

| | | |
|---|---|---|
| (ReLU, Tanh) | 0.9938 | 0.0014 |
| (Sigmoid, ReLU) | 0.9890 | 0.0027 |
| (Sigmoid, Sigmoid) | 0.9927 | 0.0016 |
| (Sigmoid, Tanh) | 0.9891 | 0.0034 |
| (Tanh, ReLU) | 0.9715 | 0.0072 |
| (Tanh, Sigmoid) | 0.9185 | 0.0310 |
| (Tanh, Tanh) | 0.9738 | 0.0055 |

7. Filter/Kernel Size

| **Hyperparameter Value (Conv, MLP)** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 3 | 0.9944 | 0.0013 |
| 5 | 0.9946 | 0.0016 |
| 7 | **0.9960** | 0.0011 |
| 9 | 0.9958 | 0.0016 |
| 11 | 0.9958 | 0.0014 |
| 13 | 0.9955 | 0.0014 |
| 15 | 0.9959 | 0.0012 |
| 17 | 0.9952 | 0.0014 |
| 19 | 0.9958 | 0.0021 |

8. Optimizer

| **Hyperparameter Value** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| SGD | **0.9956** | 0.0016 |
| RMSprop | 0.9934 | 0.0020 |
| Adam | 0.9950 | 0.0012 |
| Adamax | 0.9953 | 0.0016 |
| Nadam | 0.9950 | 0.0010 |
| Adadelta | 0.9746 | 0.0034 |

9. Loss Function

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| Sparse Categorical Cross-Entropy | **0.9945** | 0.0021 |
| Poisson | 0.4016 | 0.4016 |
| KL Divergence | 0.1606 | 0.3212 |

10. Dropout Rate

| Hyperparameter Value (Conv, MLP) | F1 Score | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| (0.1, 0.1) | 0.9932 | 0.0020 |
| (0.1, 0.3) | 0.9942 | 0.0016 |
| (0.1, 0.5) | 0.9947 | 0.0021 |
| (0.1, 0.7) | 0.9941 | 0.0015 |
| (0.1, 0.9) | 0.9923 | 0.0025 |
| (0.3, 0.1) | 0.9939 | 0.0018 |
| (0.3, 0.3) | 0.9944 | 0.0016 |
| (0.3, 0.5) | 0.9947 | 0.0017 |
| (0.3, 0.7) | 0.9947 | 0.0019 |
| (0.3, 0.9) | 0.9920 | 0.0024 |
| (0.5, 0.1) | **0.9950** | 0.0017 |
| (0.5, 0.3) | 0.9947 | 0.0019 |
| (0.5, 0.5) | 0.9945 | 0.0016 |
| (0.5, 0.7) | 0.9947 | 0.0020 |
| (0.5, 0.9) | 0.9922 | 0.0016 |
| (0.7, 0.1) | 0.9941 | 0.0013 |
| (0.7, 0.3) | 0.9945 | 0.0015 |
| (0.7, 0.5) | 0.9945 | 0.0016 |

| | | |
|---|---|---|
| (0.7, 0.7) | 0.9942 | 0.0014 |
| (0.7, 0.9) | 0.9898 | 0.0030 |
| (0.9, 0.1) | 0.9930 | 0.0019 |
| (0.9, 0.3) | 0.9932 | 0.0021 |
| (0.9, 0.5) | 0.9918 | 0.0019 |
| (0.9, 0.7) | 0.9896 | 0.0025 |
| (0.9, 0.9) | 0.9781 | 0.0053 |

11. Number of MLP Neurons

| **Hyperparameter Value** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 10 | 0.9903 | 0.0023 |
| 50 | 0.9942 | 0.0019 |
| 100 | 0.9947 | 0.0017 |
| 250 | 0.9951 | 0.0013 |
| 500 | **0.9954** | 0.0011 |
| 750 | 0.9951 | 0.0010 |
| 1000 | 0.9953 | 0.0013 |

**B. Individual Hyperparameter Search Results – Dynamic Respiratory Volume**

1. Number of Convolution Layers

| **Hyperparameter Value** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 1 | **0.8052** | 0.0099 |
| 2 | 0.7996 | 0.0097 |
| 3 | 0.8051 | 0.0115 |
| 4 | 0.7957 | 0.0144 |
| 5 | 0.7849 | 0.0179 |

2. Number of MLP Layers

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 1 | **0.8034** | 0.0132 |
| 2 | 0.7863 | 0.0178 |
| 3 | 0.7938 | 0.0137 |
| 4 | 0.7878 | 0.0169 |
| 5 | 0.7922 | 0.0128 |

3. Number of Convolution Filters

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 2 | 0.7095 | 0.0358 |
| 4 | 0.7727 | 0.0178 |
| 8 | 0.7906 | 0.0156 |
| 16 | **0.8019** | 0.0195 |
| 32 | 0.7922 | 0.0295 |
| 64 | 0.7814 | 0.0197 |
| 128 | 0.7784 | 0.0408 |
| 256 | 0.7858 | 0.0164 |
| 512 | 0.7816 | 0.0213 |
| 1024 | 0.7746 | 0.0371 |

4. Batch Size

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 32 | 0.8089 | 0.0130 |
| 64 | **0.8159** | 0.0110 |

| | | |
|---|---|---|
| 128 | 0.7997 | 0.0159 |
| 256 | 0.7958 | 0.0189 |

5. Number of Epochs

| **Hyperparameter Value** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 50 | 0.7805 | 0.0161 |
| 100 | 0.8048 | 0.0104 |
| 150 | 0.7902 | 0.0207 |
| 200 | 0.8031 | 0.0157 |
| 250 | 0.8045 | 0.0155 |
| 300 | **0.8071** | 0.0175 |

6. Activation Function

| **Hyperparameter Value (Conv, MLP)** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| (ReLU, ReLU) | **0.8098** | 0.0104 |
| (ReLU, Sigmoid) | 0.8072 | 0.0149 |
| (ReLU, Tanh) | 0.8001 | 0.0145 |
| (Sigmoid, ReLU) | 0.5770 | 0.2911 |
| (Sigmoid, Sigmoid) | 0.7869 | 0.0187 |
| (Sigmoid, Tanh) | 0.3492 | 0.3533 |
| (Tanh, ReLU) | 0.6651 | 0.0359 |
| (Tanh, Sigmoid) | 0.5035 | 0.1258 |
| (Tanh, Tanh) | 0.6339 | 0.0283 |

7. Filter/Kernel Size

| **Hyperparameter Value (Conv, MLP)** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |

| | | |
|---|---|---|
| 3 | 0.8029 | 0.0171 |
| 5 | 0.8028 | 0.0129 |
| 7 | 0.8112 | 0.0119 |
| 9 | 0.8122 | 0.0085 |
| 11 | 0.8090 | 0.0151 |
| 13 | 0.8115 | 0.0186 |
| 15 | **0.8144** | 0.0085 |
| 17 | 0.8076 | 0.0222 |
| 19 | 0.8058 | 0.0130 |

8. Optimizer

| **Hyperparameter Value** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| SGD | **0.8098** | 0.0093 |
| RMSprop | 0.7867 | 0.0104 |
| Adam | 0.8036 | 0.0143 |
| Adamax | 0.8073 | 0.0106 |
| Nadam | 0.8038 | 0.0097 |
| Adadelta | 0.7141 | 0.0120 |

9. Loss Function

| **Hyperparameter Value** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| Sparse Categorical Cross-Entropy | **0.8048** | 0.0128 |
| Poisson | 0.1227 | 0.2456 |
| KL Divergence | 0.0619 | 0.1857 |

10. Dropout Rate

| **Hyperparameter Value** | **F1 Score** |
|---|---|

| (Conv, MLP) | Average | Standard Deviation |
|---|---|---|
| (0.1, 0.1) | 0.7771 | 0.0114 |
| (0.1, 0.3) | 0.7843 | 0.0155 |
| (0.1, 0.5) | 0.7897 | 0.0111 |
| (0.1, 0.7) | 0.7903 | 0.0103 |
| (0.1, 0.9) | 0.7586 | 0.0146 |
| (0.3, 0.1) | 0.7901 | 0.0109 |
| (0.3, 0.3) | 0.7931 | 0.0141 |
| (0.3, 0.5) | 0.7980 | 0.0113 |
| (0.3, 0.7) | 0.7972 | 0.0152 |
| (0.3, 0.9) | 0.7338 | 0.0323 |
| (0.5, 0.1) | 0.7935 | 0.0129 |
| (0.5, 0.3) | 0.8030 | 0.0111 |
| (0.5, 0.5) | **0.8131** | 0.0110 |
| (0.5, 0.7) | 0.7965 | 0.0186 |
| (0.5, 0.9) | 0.7348 | 0.0421 |
| (0.7, 0.1) | 0.7914 | 0.0188 |
| (0.7, 0.3) | 0.7974 | 0.0195 |
| (0.7, 0.5) | 0.8052 | 0.0188 |
| (0.7, 0.7) | 0.8035 | 0.0159 |
| (0.7, 0.9) | 0.7206 | 0.0446 |
| (0.9, 0.1) | 0.7607 | 0.0275 |
| (0.9, 0.3) | 0.7580 | 0.0183 |
| (0.9, 0.5) | 0.7560 | 0.0327 |
| (0.9, 0.7) | 0.6984 | 0.0313 |
| (0.9, 0.9) | 0.5975 | 0.0609 |

11. Number of MLP Neurons

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | Average | Standard Deviation |

| | | |
|---|---|---|
| 10 | 0.7679 | 0.0245 |
| 50 | 0.7993 | 0.0166 |
| 100 | 0.8033 | 0.0149 |
| 250 | 0.8071 | 0.0121 |
| 500 | **0.8123** | 0.0084 |
| 750 | 0.8121 | 0.0098 |
| 1000 | 0.8121 | 0.0105 |

### C. Individual Hyperparameter Search Results – Dynamic Respiratory Phase

1. Number of Convolution Layers

| **Hyperparameter Value** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 1 | 0.8371 | 0.0102 |
| 2 | **0.8410** | 0.0068 |
| 3 | 0.8380 | 0.0081 |
| 4 | 0.8391 | 0.0086 |
| 5 | 0.8370 | 0.0070 |

2. Number of MLP Layers

| **Hyperparameter Value** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 1 | **0.8385** | 0.0077 |
| 2 | 0.8274 | 0.0085 |
| 3 | 0.8331 | 0.0085 |
| 4 | 0.8300 | 0.0103 |
| 5 | 0.8353 | 0.0090 |

3. Number of Convolution Filters

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | Average | Standard Deviation |
| 2 | 0.7857 | 0.0255 |
| 4 | 0.8205 | 0.0145 |
| 8 | 0.8394 | 0.0084 |
| 16 | **0.8429** | 0.0079 |
| 32 | 0.8352 | 0.0123 |
| 64 | 0.8307 | 0.0109 |
| 128 | 0.8275 | 0.0145 |
| 256 | 0.8241 | 0.0090 |
| 512 | 0.8200 | 0.0131 |
| 1024 | 0.8194 | 0.0103 |

4. Batch Size

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | Average | Standard Deviation |
| 32 | **0.8409** | 0.0075 |
| 64 | 0.8345 | 0.0116 |
| 128 | 0.8375 | 0.0077 |
| 256 | 0.8346 | 0.0109 |

5. Number of Epochs

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | Average | Standard Deviation |
| 50 | 0.8324 | 0.0087 |
| 100 | 0.8376 | 0.0101 |
| 150 | 0.8396 | 0.0057 |
| 200 | 0.8379 | 0.0111 |
| 250 | 0.8369 | 0.0071 |
| 300 | **0.8410** | 0.0117 |

6. Activation Function

| Hyperparameter Value (Conv, MLP) | F1 Score | |
|---|---|---|
| | Average | Standard Deviation |
| (ReLU, ReLU) | 0.8426 | 0.0093 |
| (ReLU, Sigmoid) | **0.8459** | 0.0092 |
| (ReLU, Tanh) | 0.8417 | 0.0066 |
| (Sigmoid, ReLU) | 0.7791 | 0.0234 |
| (Sigmoid, Sigmoid) | 0.8389 | 0.0093 |
| (Sigmoid, Tanh) | 0.7723 | 0.0562 |
| (Tanh, ReLU) | 0.8232 | 0.0119 |
| (Tanh, Sigmoid) | 0.7928 | 0.0157 |
| (Tanh, Tanh) | 0.8020 | 0.0182 |

7. Filter/Kernel Size

| Hyperparameter Value (Conv, MLP) | F1 Score | |
|---|---|---|
| | Average | Standard Deviation |
| 3 | 0.8416 | 0.0107 |
| 5 | 0.8407 | 0.0070 |
| 7 | **0.8430** | 0.0127 |
| 9 | 0.8380 | 0.0077 |
| 11 | 0.8388 | 0.0098 |
| 13 | 0.8413 | 0.0054 |
| 15 | 0.8402 | 0.0063 |
| 17 | 0.8319 | 0.0103 |
| 19 | 0.8389 | 0.0093 |

8. Optimizer

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| SGD | 0.8449 | 0.0074 |
| RMSprop | 0.8233 | 0.0169 |
| Adam | 0.8406 | 0.0070 |
| Adamax | 0.8454 | 0.0084 |
| Nadam | **0.8476** | 0.0054 |
| Adadelta | 0.7628 | 0.0121 |

9. Loss Function

| Hyperparameter Value | F1 Score | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| Sparse Categorical Cross-Entropy | **0.8435** | 0.0072 |
| Poisson | 0.2035 | 0.3109 |
| KL Divergence | 0.0685 | 0.2055 |

10. Dropout Rate

| Hyperparameter Value (Conv, MLP) | F1 Score | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| (0.1, 0.1) | 0.8268 | 0.0096 |
| (0.1, 0.3) | 0.8340 | 0.0081 |
| (0.1, 0.5) | 0.8350 | 0.0080 |
| (0.1, 0.7) | 0.8347 | 0.0077 |
| (0.1, 0.9) | 0.8011 | 0.0146 |
| (0.3, 0.1) | 0.8324 | 0.0101 |
| (0.3, 0.3) | 0.8340 | 0.0089 |
| (0.3, 0.5) | 0.8373 | 0.0111 |
| (0.3, 0.7) | 0.8376 | 0.0083 |
| (0.3, 0.9) | 0.7682 | 0.0181 |

| | | |
|---|---|---|
| (0.5, 0.1) | 0.8387 | 0.0106 |
| (0.5, 0.3) | **0.8415** | 0.0076 |
| (0.5, 0.5) | 0.8384 | 0.0076 |
| (0.5, 0.7) | 0.8394 | 0.0121 |
| (0.5, 0.9) | 0.8069 | 0.0144 |
| (0.7, 0.1) | 0.8356 | 0.0095 |
| (0.7, 0.3) | 0.8372 | 0.0106 |
| (0.7, 0.5) | 0.8328 | 0.0078 |
| (0.7, 0.7) | 0.8357 | 0.0103 |
| (0.7, 0.9) | 0.7983 | 0.0176 |
| (0.9, 0.1) | 0.8175 | 0.0143 |
| (0.9, 0.3) | 0.8052 | 0.0193 |
| (0.9, 0.5) | 0.7874 | 0.0211 |
| (0.9, 0.7) | 0.7624 | 0.0273 |
| (0.9, 0.9) | 0.6950 | 0.0215 |

11. Number of MLP Neurons

| **Hyperparameter Value** | **F1 Score** | |
|---|---|---|
| | **Average** | **Standard Deviation** |
| 10 | 0.8119 | 0.0161 |
| 50 | 0.8274 | 0.0108 |
| 100 | 0.8419 | 0.0101 |
| 250 | 0.8413 | 0.0084 |
| 500 | 0.8453 | 0.0086 |
| 750 | **0.8471** | 0.0035 |
| 1000 | 0.8465 | 0.0089 |

# VII. REFERENCES


[1] G. A. Roth *et al.*, "Global burden of cardiovascular diseases and risk factors, 1990–2019: update from the GBD 2019 study," vol. 76, no. 25, pp. 2982-3021, 2020.

[2] C. W. Tsao *et al.*, "Heart disease and stroke statistics—2022 update: a report from the American Heart Association," vol. 145, no. 8, pp. e153-e639, 2022.

[3] D. Bloom *et al.*, "The global economic burden of noncommunicable diseases. Geneva: World Economic Forum; 2011," *Available from: URL: http://www.hsph.harvard.edu/pgda/working.htm,* 2016.

[4] Z. Jin, Y. Sun, and A. C. Cheng, "Predicting cardiovascular disease from real-time electrocardiographic monitoring: An adaptive machine learning approach on a cell phone," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 6889-6892: IEEE.

[5] S. Giancaterino, F. Lupercio, M. Nishimura, and J. C. Hsu, "Current and future use of insertable cardiac monitors," *JACC: Clinical Electrophysiology,* vol. 4, no. 11, pp. 1383-1396, 2018.

[6] R. A. Bernstein *et al.*, "Effect of long-term continuous cardiac monitoring vs usual care on detection of atrial fibrillation in patients with stroke attributed to large-or small-vessel disease: the STROKE-AF randomized clinical trial," vol. 325, no. 21, pp. 2169-2177, 2021.

[7] S. F. Weng, J. Reps, J. Kai, J. M. Garibaldi, and N. Qureshi, "Can machine-learning improve cardiovascular risk prediction using routine clinical data?," *PloS one,* vol. 12, no. 4, p. e0174944, 2017.

[8] N. Mandal, "Respirometers including spirometer, pneumotachograph and peak flow meter," *Anaesthesia Intensive Care Medicine,* vol. 7, no. 1, pp. 1-5, 2006.

[9] J. W. Kreit and F. C. Sciurba, "The accuracy of pneumotachograph measurements during mechanical ventilation," *American Journal of Respiratory Critical Care Medicine,* vol. 154, no. 4, pp. 913-917, 1996.

[10] A. Hobbes, "A comparison of methods of calibrating the pneumotachograph," *British Journal of Anaesthesia,* vol. 39, no. 11, pp. 899-907, 1967.

[11] M. R. Miller, S. A. Dickinson, and D. J. Hitchings, "The accuracy of portable peak flow meters," vol. 47, no. 11, pp. 904-909, 1992.

[12] G. K. Wolf and J. H. Arnold, "Noninvasive assessment of lung volume: respiratory inductance plethysmography and electrical impedance tomography," *Critical Care Medicine,* vol. 33, no. 3, pp. S163-S169, 2005.

[13] J. Stradling, G. Chadwick, C. Quirk, and T. Phillips, "Respiratory inductance plethysmography: calibration techniques, their validation and the effects of posture," *Bulletin Europeen de Physiopathologie Respiratoire,* vol. 21, no. 4, pp. 317-324, 1985.

[14] H. Qudsi and M. Gupta, "Low-cost, thermistor based respiration monitor," in *2013 29th Southern Biomedical Engineering Conference*, 2013, pp. 23-24: IEEE.

[15] C. Xiong, B. J. Sjöberg, P. Sveider, P. Ask, D. Loyd, and B. Wranne, "Problems in timing of respiration with the nasal thermistor technique," *Journal of the American Society of Echocardiography,* vol. 6, no. 2, pp. 210-216, 1993.

[16] M. AlGhatrif and J. Lindsay, "A brief review: history to understand fundamentals of electrocardiography," *Journal of Community Hospital Internal Medicine Perspectives,* vol. 2, no. 1, p. 14383, 2012.

[17] G. S. Wagner, *Marriott's Practical Electrocardiography*. Lippincott Williams & Wilkins, 2001.

[18] A. N. Sharma and A. Baranchuk, "Ambulatory External Electrocardiography Monitoring: Holter, Extended Holter, Mobile Cardiac Telemetry Monitoring," *Cardiac Electrophysiology Clinics,* vol. 13, no. 3, pp. 427-438, 2021.

[19] J. Achten and A. E. Jeukendrup, "Heart rate monitoring," *Sports Medicine,* vol. 33, no. 7, pp. 517-538, 2003.

[20] G. K. Prasad and J. Sahambi, "Classification of ECG arrhythmias using multi-resolution analysis and neural networks," in *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, 2003, vol. 1, pp. 227-231: IEEE.

[21] T. Stamkopoulos, K. Diamantaras, N. Maglaveras, and M. Strintzis, "ECG analysis using nonlinear PCA neural networks for ischemia detection," *IEEE Transactions on Signal Processing,* vol. 46, no. 11, pp. 3058-3067, 1998.

[22] E. O. Nishime, C. R. Cole, E. H. Blackstone, F. J. Pashkow, and M. S. Lauer, "Heart rate recovery and treadmill exercise score as predictors of mortality in patients referred for exercise ECG," vol. 284, no. 11, pp. 1392-1398, 2000.

[23] A. Sherwood, M. T. Allen, J. Fahrenberg, R. M. Kelsey, W. R. Lovallo, and L. J. Van Doornen, "Methodological guidelines for impedance cardiography," *Psychophysiology,* vol. 27, no. 1, pp. 1-23, 1990.

[24] M. Qu, Y. Zhang, J. G. Webster, and W. J. Tompkins, "Motion artifact from spot and band electrodes during impedance cardiography," *IEEE Transactions on Biomedical Engineering,* no. 11, pp. 1029-1036, 1986.

[25] M. Nitzan, A. Romem, and R. Koppel, "Pulse oximetry: fundamentals and technology update," *Medical Devices,* vol. 7, p. 231, 2014.

[26] B. S. Kim and S. K. Yoo, "Motion artifact reduction in photoplethysmography using independent component analysis," *IEEE Transactions on Biomedical Engineering,* vol. 53, no. 3, pp. 566-568, 2006.

[27] G. Lu, F. Yang, J. Taylor, and J. Stein, "A comparison of photoplethysmography and ECG recording to analyse heart rate variability in healthy subjects," *Journal of Medical Engineering Technology,* vol. 33, no. 8, pp. 634-641, 2009.

[28] S. Whiting, S. Moreland, J. Costello, G. Colopy, and C. McCann, "Recognising cardiac abnormalities in wearable device photoplethysmography (PPG) with deep learning," 2018.

[29] Q. Yousef, M. Reaz, and M. A. M. Ali, "The analysis of PPG morphology: investigating the effects of aging on arterial compliance," *Measurement Science Review,* vol. 12, no. 6, p. 266, 2012.

[30] J. S. Gottdiener *et al.*, "American Society of Echocardiography recommendations for use of echocardiography in clinical trials: a report from the american society of echocardiography's guidelines and standards committee and the task force on echocardiography in clinical trials," *Journal of the American Society of Echocardiography,* vol. 17, no. 10, pp. 1086-1119, 2004.

[31] J. Bogaert, S. Dymarkowski, A. M. Taylor, and V. Muthurangu, *Clinical cardiac MRI*. Springer Science & Business Media, 2012.

[32] I. C. Jeong and J. Finkelstein, "Introducing contactless assessment of heart rate variability using high speed video camera," in *2015 36th IEEE Sarnoff Symposium*, 2015, pp. 7-11: IEEE.

[33] O. T. Inan *et al.*, "Ballistocardiography and seismocardiography: A review of recent advances," *IEEE Journal of Biomedical and Health Informatics,* vol. 19, no. 4, pp. 1414-1427, 2014.

[34] J. Gordon, "Certain molar movements of the human body produced by the circulation of the blood," *Journal of Anatomy & Physiology,* vol. 11, no. Pt 3, p. 533, 1877.

[35] I. Starr, A. Rawson, and H. Schroeder, "Apparatus for recording the heart's recoil and the blood's impacts in man (ballistocardiograph), experiments on the principles involved, records in normal and abnormal conditions," in *Proceedings of the Am. Physiol. SOC. 50th Annual Meeting. Am. J. Physiology*, 1938, vol. 123, p. 195.

[36] W. Tucker, "The Hot Wire Microphone and its application to Problems of Sound," *RSA Journal,* vol. 71, no. 3659, pp. 121-137, 1923.

[37] C. Heald and W. Tucker, "Recoil curves as shown by the hot-wire microphone," *Proceedings of the Royal Society of London. Series B, Containing Papers of a Biological Character,* vol. 93, no. 652, pp. 281-298, 1922.

[38] A. Taebi, B. E. Solar, A. J. Bomar, R. H. Sandler, and H. A. Mansy, "Recent Advances in Seismocardiography," *Vibration,* vol. 2, no. 1, pp. 64-86, 2019.

[39] D. M. Salerno, "Seismocardiography: A new technique for recording cardiac vibrations. concept, method, and initial observations," *Journal of Cardiovascular Technology,* vol. 9, no. 2, pp. 111-118, 1990.
[40] Y. D'Mello *et al.*, "Autocorrelated differential algorithm for real-time seismocardiography analysis," vol. 19, no. 13, pp. 5127-5140, 2019.
[41] P. H. Charlton *et al.*, "Breathing rate estimation from the electrocardiogram and photoplethysmogram: A review," *IEEE Reviews in Biomedical Engineering,* vol. 11, pp. 2-20, 2017.
[42] D. M. Quarrie and P. Neary, "Comparison of seismocardiography to echocardiography for measuring cardiac cycle events," 2011.
[43] R. S. Crow, P. Hannan, D. Jacobs, L. Hedquist, and D. M. Salerno, "Relationship between seismocardiogram and echocardiogram for events in the cardiac cycle," *American Journal of Noninvasive Cardiology,* vol. 8, pp. 39-46, 1994.
[44] K. Sørensen, S. E. Schmidt, A. S. Jensen, P. Søgaard, and J. J. Struijk, "Definition of fiducial points in the normal seismocardiogram," *Scientific Reports,* vol. 8, no. 1, pp. 1-11, 2018.
[45] K. Tavakolian, "Characterization and analysis of seismocardiogram for estimation of hemodynamic parameters," Applied Science: School of Engineering Science, 2010.
[46] O. T. Inan *et al.*, "Ballistocardiography and seismocardiography: A review of recent advances," vol. 19, no. 4, pp. 1414-1427, 2014.
[47] Y. D'Mello *et al.*, "Real-Time Cardiac Beat Detection and Heart Rate Monitoring from Combined Seismocardiography and Gyrocardiography," *Sensors,* vol. 19, no. 16, p. 3472, 2019.
[48] M. Di Rienzo *et al.*, "Wearable seismocardiography: Towards a beat-by-beat assessment of cardiac mechanics in ambulant subjects," vol. 178, no. 1-2, pp. 50-59, 2013.
[49] A. Bricout, J. Fontecave-Jallon, D. Colas, G. Gerard, J.-L. Pépin, and P.-Y. Guméry, "Adaptive accelerometry derived respiration: Comparison with respiratory inductance plethysmography during sleep," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 6714-6717: IEEE.
[50] S. Transue, P. Nguyen, T. Vu, and M.-H. Choi, "Real-time tidal volume estimation using iso-surface reconstruction," in *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, 2016, pp. 209-218: IEEE.
[51] M. Chu *et al.*, "Respiration rate and volume measurements using wearable strain sensors," *NPJ Digital Medicine,* vol. 2, no. 1, pp. 1-9, 2019.
[52] A. Shahshahani, C. Laverdiere, S. Bhadra, and Z. J. S. Zilic, "Ultrasound sensors for diaphragm motion tracking: An application in non-invasive respiratory monitoring," vol. 18, no. 8, p. 2617, 2018.
[53] Y. S. Lee, P. N. Pathirana, C. L. Steinfort, and T. Caelli, "Monitoring and analysis of respiratory patterns using microwave doppler radar," *IEEE Journal of Translational Engineering in Health Medicine,* vol. 2, pp. 1-12, 2014.
[54] M. J. Tadi *et al.*, "Gyrocardiography: A new non-invasive approach in the study of mechanical motions of the heart. Concept, method and initial observations," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 2034-2037: IEEE.
[55] P.-F. Migeotte, V. Mucci, Q. Delière, L. Lejeune, and P. Van De Borne, "Multi-dimensional kineticardiography a new approach for wearable cardiac monitoring through body acceleration recordings," in *XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016*, 2016, pp. 1125-1130: Springer.
[56] J. Skoric *et al.*, "Respiratory Modulation of Sternal Motion in the Context of Seismocardiography," vol. 22, no. 13, pp. 13055-13066, 2022.
[57] A. Akhbardeh, S. Junnila, M. Koivuluoma, T. Koivistoinen, and A. Varri, "The heart disease diagnosing system based on force sensitive chair's measurement, biorthogonal wavelets and

neural networks," in *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, 2005, pp. 676-681: IEEE.
[58] N. Clairmonte *et al.*, "Neural Network-based Classification of Static Lung Volume States using Vibrational Cardiography," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2020, pp. 221-224: IEEE.
[59] W. J. Walker, "Ballistocardiography in Cardiovascular Research," *Archives of Internal Medicine,* vol. 121, no. 4, pp. 387-387, 1968.
[60] K. Tavakolian, A. Vaseghi, and B. Kaminska, "Improvement of ballistocardiogram processing by inclusion of respiration information," *Physiological Measurement,* vol. 29, no. 7, p. 771, 2008.
[61] A. Taebi and H. A. Mansy, "Grouping similar seismocardiographic signals using respiratory information," in *2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2017, pp. 1-6: IEEE.
[62] B. E. Solar, A. Taebi, and H. A. Mansy, "Classification of seismocardiographic cycles into lung volume phases," in *2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2017, pp. 1-2: IEEE.
[63] M. J. Tadi, T. Koivisto, M. Pänkäälä, and A. Paasio, "Accelerometer-based method for extracting respiratory and cardiac gating information for dual gating during nuclear medicine imaging," *Journal of Biomedical Imaging,* vol. 2014, p. 6, 2014.
[64] V. Zakeri, A. Akhbardeh, N. Alamdari, R. Fazel-Rezai, M. Paukkunen, and K. Tavakolian, "Analyzing seismocardiogram cycles to identify the respiratory phases," *IEEE Transactions on Biomedical Engineering,* vol. 64, no. 8, pp. 1786-1792, 2016.
[65] J. Skoric, Y. D'Mello, M. Lortie, S. Gagnon, and D. V. Plant, "Effect of Static Respiratory Volume on the Waveform of Cardiac-induced Sternal Vibrations," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 4917-4921: IEEE.
[66] R. M. Rangayyan, *Biomedical signal analysis*. John Wiley & Sons, 2015.
[67] K. R. Foster, R. Koprowski, and J. D. Skufca, "Machine learning, medical diagnosis, and biomedical engineering research-commentary," *Biomedical Engineering Online,* vol. 13, no. 1, pp. 1-9, 2014.
[68] C. M. Biship, "Pattern recognition and machine learning (information science and statistics)," ed: Springer New York, 2007.
[69] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.
[70] S. B. Kotsiantis, "Decision trees: a recent overview," *Artificial Intelligence Review,* vol. 39, no. 4, pp. 261-283, 2013.
[71] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning,* vol. 20, no. 3, pp. 273-297, 1995.
[72] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2021.
[73] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems,* vol. 9, 1996.
[74] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets,* vol. 31, no. 3, pp. 685-695, 2021.
[75] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
[76] Y. LeCun, Y. Bengio, and G. J. n. Hinton, "Deep learning," vol. 521, no. 7553, pp. 436-444, 2015.
[77] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature,* vol. 323, no. 6088, pp. 533-536, 1986.
[78] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[79] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, 2017, pp. 1-6: IEEE.
[80] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278-2324, 1998.
[81] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mechanical Systems & Signal Processing,* vol. 151, p. 107398, 2021.
[82] W. Zhang, G. Peng, and C. Li, "Bearings fault diagnosis based on convolutional neural networks with 2-D representation of vibration signals as input," in *MATEC web of conferences*, 2017, vol. 95, p. 13001: EDP Sciences.
[83] J. T. Ruiz, J. D. B. Pérez, and J. R. B. Blázquez, "Arrhythmia detection using convolutional neural models," in *International Symposium on Distributed Computing and Artificial Intelligence*, 2018, pp. 120-127: Springer.
[84] S. Kiranyaz, T. Ince, R. Hamila, and M. Gabbouj, "Convolutional neural networks for patient-specific ECG classification," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 2608-2611: IEEE.
[85] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," *IEEE Transactions on Biomedical Engineering,* vol. 63, no. 3, pp. 664-675, 2015.
[86] O. Abdeljaber, S. Sassi, O. Avci, S. Kiranyaz, A. A. Ibrahim, and M. Gabbouj, "Fault detection and severity identification of ball bearings by online condition monitoring," *IEEE Transactions on Industrial Electronics,* vol. 66, no. 10, pp. 8136-8147, 2018.
[87] L. Eren, T. Ince, and S. Kiranyaz, "A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier," *Journal of Signal Processing Systems,* vol. 91, no. 2, pp. 179-189, 2019.
[88] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Transactions on Industrial Electronics,* vol. 63, no. 11, pp. 7067-7075, 2016.
[89] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
[90] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Science,* vol. 6, no. 12, pp. 310-316, 2017.
[91] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807-814.
[92] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.
[93] M. C. Mukkamala and M. Hein, "Variants of rmsprop and adagrad with logarithmic regret bounds," in *International conference on machine learning*, 2017, pp. 2545-2553: PMLR.
[94] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139-1147: PMLR.
[95] InvenSense, "MPU-9250 Product Specification," InvenSense, Ed., ed, 2018.
[96] F. N. Wilson, F. D. Johnston, F. F. Rosenbaum, and P. S. Barker, "On Einthoven's triangle, the theory of unipolar electrocardiographic leads, and the interpretation of the precordial electrocardiogram," 1946.
[97] G. vanRossum, "Python reference manual," *Department of Computer Science,* no. R 9525, 1995.
[98] M. Grinberg, *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.", 2018.

[99] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering,* no. 3, pp. 230-236, 1985.
[100] J. D. Kelleher, B. Mac Namee, and A. D'arcy, *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press, 2020.
[101] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory & Neural Networks,* vol. 3361, no. 10, p. 1995, 1995.
[102] Y. Kaya, S. Hong, and T. Dumitras, "Shallow-deep networks: Understanding and mitigating network overthinking," in *International conference on machine learning*, 2019, pp. 3301-3310: PMLR.
[103] C. W. Wu, "ProdSumNet: Reducing Model Parameters in Deep Neural Networks via Poduct-of-Sums Matrix Decompositions," 2018.