An IoT, Sensor-Based, Non-Invasive Dog Training Programme Assistant

by

Justas Cepaitis

This thesis has been submitted in partial fulfillment for the degree of Bachelor of Science in Software Development

in the
Faculty of Engineering and Science
Department of Computer Science

May 2022

Declaration of Authorship

I, Justas Cepaitis , declare that this thesis titled, 'An IoT, Sensor-Based, Non-Invasive Dog Training Programme Assistant' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for an undergraduate degree at Munster Technological University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at Munster Technological University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project report is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Justas Cepaitis

Date: 04/05/2022

Munster Technological University

Abstract

Faculty of Engineering and Science
Department of Computer Science

Bachelor of Science

by Justas Cepaitis

Understanding the nature of a dog physiology and temperament is imperative for the owner or caretaker when it comes to managing training sessions, monitoring pet's physical health, and, in general, turning the dog-human relationship into a healthier and more empathic one.

This project aims to provide an automated system for monitoring certain aspects of dog's health and behaviour. Specifically, the goal is to provide a system able to monitor dog's engagements with its surroundings by correlating its location with its physiological and psychological state through a health monitoring device. By doing so, it will be possible to identify and catalogue moments of stress and/or excitement for the dog. Whereas the above will be valuable in average pet ownership scenarios, it becomes particularly appealing to assist trainers and volunteering foster families in dogs training programmes. With this in mind, the project contributions have been discussed with the NGO - Autism Assistance Dogs Ireland (AADI), for the project to cater their training programmes as a real-world use-case application.

First, the project will monitor the geolocation of the dog, to identify its location over time for managing ambulatory activities with training conditions, such as pet walks. This location information will be combined with the monitoring of physical health factors (such as the heart rate variability) using a health monitoring system. The combined information will be used to infer stress/excitement levels of the dog, and correlate them to specific situations faced during training. The project goals will be achieved by leveraging an IoT, sensor-based, non-invasive system mounted on the dog, that will be made remotely accessible through a mobile application.

Acknowledgements

Ignacio Castiñeiras, Thank You!
I would like to dedicate a genuine and personal *Thank You* to Ignacio Castiñeiras for his candour and genuine character when guiding me through the initial Research Phase of this Final Year Project. He was an imperative cog in the wheel of operation of this project in guiding me along as I lost my way every now and then. He kept me on track when it was easy for me to lose my way.

Autism Assistance Dogs Ireland (AADI), Thank You!
I would like to dedicate another personal *Thank You!* to AADI for taking the time out of their busy days to make their contribution in helping envision the concept and primarily the purpose this project may serve. Their word had not gone unnoticed and their participation made me realize just how potentially useful the prospects of such a system may be, particularly in the induction of new guide dogs, which the world sorely needs to aid the more vulnerable people in ours, and others' communities.

Victor Cionca, Thank You!
I would like to dedicate a genuine and personal *Thank You* to Victor Cionca for accomodating me during my second phase of the Final Year Project, namely the implementation phase, which I would be lying if I said was not fraught with challenges that caused me to go off track a number of times. Victor helped me keep my head steady and keep my work steadfast and focused by guiding me along and most importantly to me at least, giving me a lot of positive reinforcement that truthfully bolstered my initiative significantly.

Contents

List of Figures

List of Tables

Abbreviations

IOT	**I**nternet **O**f **T**hings
HRV	**H**eart **R**ate **V**ariability
GPS	**G**lobal **P**ositioning **S**ystem
HR	**H**eart **R**ate
ANS	**A**utonomic **N**ervous **S**ystem
GNSS	**G**lobal **N**avigation **S**atellite **S**ystem
SIM	**S**ubscriber **I**dentification **M**odule
GSM	**G**lobal **S**ystem for **M**obile Communications
GPRS	**G**eneral **P**acket **R**adio **S**ervice
TCP	**T**ransmission **C**ontrol **P**rotocol
IP	**I**nternet **P**rotocol
AADI	**A**utism **A**ssistance **D**ogs **I**reland
SDNN	**S**standard **D**eviation of **NN** intervals
RMSSD	**R**oot **M**ean **S**quare of **S**uccessive **D**ifferences
RRI	**R-R I**ntervals
HF	**H**igh **F**requency
LF	**L**ow **F**requency
IBI	**I**nter **B**eat **I**ntervals
API	**A**pplication **P**rogramming **I**nterface
ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
UID	**U**nique **I**dentifier
SMS	**S**hort **M**essage **S**ervice
QOS	**Q**uality **O**f **S**ervice

Dedicated to my mother for her undying will to help me achieve my goals...

Chapter 1

Introduction

The field of Internet of Things (IOT) has been a rapidly growing sector in computing, digitization, and automation of various manual processes. Due to its inter-operability, IoT systems are indispensable in any system that requires data collected from its real surroundings in order to operate. Various robust IoT systems such as Global Positioning System (GPS) and health monitoring systems have been designed within this space to allow for geolocation of any object, stationary or mobile from any point on the globe, and health monitoring systems have paved the way for an unprecedented level of security in terms of a healthy livelihood. From preemptive measures in micro-managing various chronic diseases ensuring persisted safety, to preventative methods that can be employed to detect and combat early signs of various ailments, IoT based systems have made possible an astonishing level of accessibility and data analysis in the complex field of biology. This, in turn, has allowed scientists to derive crucial investigations on the correlation between biological health and its effects on various facets of a living body in real-time, particularly, their emotional states.

In line with the growth of IoT, this project proposes an IoT, Sensor-based, non-invasive system that can monitor a dog's health, geolocation and derive psychological states of the animal from the health data collected. The system would employ a novel vest, rigged with GPS and heart-rate sensors that, paired with a back-end system for persisted sensor data storage and data computation capability, can output this valuable information to a mobile application remotely and in real-time.

Whereas the above will be valuable in average pet ownership scenarios, it becomes particularly appealing to assist trainers and volunteering foster families in dogs training programmes, allowing for caretakers or owners to monitor their location, and the dog's health as well as its behaviour during various interactions. The prospects of a basic vision system for further scrutiny of dog stress/excitement levels by correlating its location with its physiological and psychological state is also briefly highlighted and discussed.

1.1 Motivation

In animals, particularly domesticated ones such as dogs, it becomes rather important for the owner or caretaker to possess information that would aid them in understanding their companion at a psychological and physiological level. Knowing a dog's behaviour patterns in real-time can aid the average, unlearned owner in taking appropriate actions during training of a dog as well as during engaging activities such as walks and play-time, with or without other beings, allowing them to better moderate and navigate the activities their dog has within, and outside of public spaces. Furthermore, having a meaningful representation of the canine's current state of health and mental state will increase with great assistance to a, let's say, less informed dog owner.

Knowing a system exists by which they can identify moments of stress in the dog with relation to its engagements with their surrounding, they can interpret and, more-over, correlate/infer what may be stressful to the canine. This is something that can be of great value to any owner, but it becomes particularly appealing to assist trainers and volunteering foster families in dogs training programmes.

In general ownership situations, owners do not tend to care as much when it comes to determining the stress levels of their dog as compared to organizations that commit to training canines. In training guide dogs particularly, understanding the stress and excitement levels and being able to better interpret them is a crucial and core factor of inducting new guide dogs. A dog that is consistently stressed during training might not be a good fit as a guide dog. This process is, at times, fraught with false positives or false negatives, when the observations may have been met with an incorrect assumption. In such situations, having a system by which observations can be reinforced, particularly in the early stages of induction of guide dogs (when the canine is fostered in a home with average, untrained individuals). Moreover, a more fluid communication and a greater sense of understanding can be developed between the family fostering the dog and the trainer guiding them on what to do. With this heightened level of understanding and feedback, signs of incompatibility in a canine during training can be detected earlier, reducing further stress on the dog and, ultimately, assisting the organisation on bringing further to their programmes the dogs that are a best fit to them.

1.2 Contribution

The research into the novel implementation of the canine monitoring system was discussed in collaboration with Autism Assistance Dogs Ireland (AADI) in order to evaluate possible shortcomings of certain features of the proposed monitoring system such as the camera system for visually identifying the stressors for a canine. Their word influenced the resulting envisioning of the conceptualized canine monitoring application as a result of the system being largely inspired to aid in the training and induction of new guide dogs. The on-going discussions helped envision the system in light of more realistic scenarios.

Chapter 2

Background

2.1 Thematic Area within Computer Science

The core topic of the project is to design a resourceful and informative IoT based mobile application for monitoring canine stress levels with relation to their surroundings and potentially identify the source of their stress. Through utilizing health monitoring sensors to derive stress levels in the canine, the system can evaluate the location and possibly identify the direct source of stress for the canine during the encounter with live response. This monitoring application is intended particularly for evaluating canine performance during foster training, to evaluate their candidacy for further training. The canine monitoring application would be capable of sensing pet geolocation, health, and behaviour and therefore, in this chapter we review the state-of-the-art for the topics of Geolocation, Health Monitoring, IoT, and Mobile applications.

2.1.1 IoT Devices

IoT devices as described in [5], are systems that envelop some previously unintelligent physical object with data gathering functionalities through embedded systems (primarily sensors), and combines the feature of connectivity over the internet to inter-connect the output of the data gathering system to servers that process and store, and applications that present the newfound information to the end-user. IoT devices have been particularly inspired within the space of agriculture and animal monitoring. Information gathering devices capable of monitoring animal location within farms [6], and particularly within the space of canine monitoring, many prototype systems for capturing physical health, location and mobility monitoring, and scientific analysis of Heart Rate Variability (HRV) and its effects on the behaviour of a canine [7] has been done at a practical level [8] [9] [10]. Proven technical research identifies the resourcefulness of IoT devices utilizing embedded systems within a larger IoT infrastructure capable of canine monitoring, identifying particular information pertaining to the unique biological factors of a canine. Moreover, IoT devices encompass embedded systems and provide them connectivity with generally cloud-based solutions such as servers and data stores or applications that display the information at a higher level from the low level raw interpretation. Since these systems encompass embedded systems that usually consist of one or more sensors, they can accommodate any data-gathering piece of sensor hardware with modularity that makes them compatible for integration within an IoT system. These embedded systems also provide the built in sensors with interfaces that supply commands that handle the sensor data retrieval, and when combined with micro controllers and Integrated Development Environments (IDE) allow for limitless sensor data manipulation, pre-processing, and resource management. Furthermore, the application of IoT devices in systems that remotely transmit data is the primary example of the importance of IoT based devices in the canine monitoring application due to the wireless and remote nature of the system which stresses the necessity of remote communications between the devices and their data as well as the servers and mobile application that processes and displays the information.

2.1.2 IoT Systems

IoT systems, in contrast to IoT devices, are the end-to-end solutions that encompasses a fully functional and complete IoT infrastructure consisting of the data-gathering IoT devices, internet connectivity and protocols, and servers and devices that may involve some application(s). As described in [5], IoT systems consist of 4 key parts: IoT devices that collect the information, Gateways that derive the communication methods and protocols between the various components, Cloud that involves servers and processing/storage of data, and applications that deliver the processed information to the end user in real-time or otherwise. Figure 2.1 depicts the 4 key parts of IoT systems. This system infrastructure particularly pertains to the application of the canine monitoring system which necessitates IoT devices for collecting data related to the canine, and the canine's location, as well as broadcasting the data over communication gateways. Furthermore, the processing of this data would be necessary for any computation that is required to transform the raw data into a more meaningful representation that can be broadcasted and interpreted by the end user on the mobile application.

The fundamental design of IoT system architectures is inspired by automation which reduces the necessity of human and computation machine interactivity during operation through purely autonomous design. Readable sensor systems within an IoT device inter-connected with cloud computing over internet that can manage the sensor's resources through intelligent communication protocols effectively allows the system to inter-operate and manage its own resources without the necessity of human intervention. This level of automation is key for a canine monitoring system that can provide a real-time service for monitoring a dog without the need of human intervention, providing the owner/caretaker of the canine companion a monitoring system that *just works*.

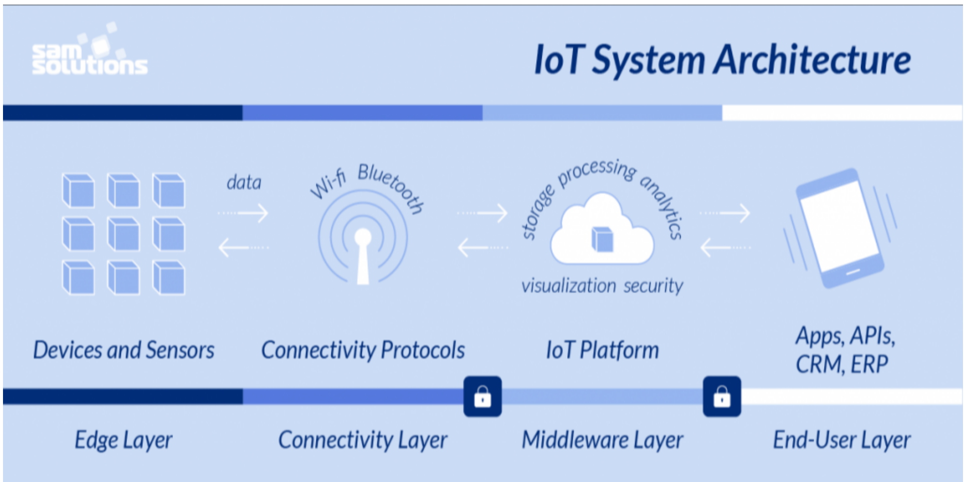


FIGURE 2.1: A simple diagram depicting the common architecture of an IoT system [1]

2.1.3 Health Monitoring Systems

Health monitoring systems are as the name suggests, systems that are capable of monitoring various facets of physical health of a given body. The emphasis in the space of health monitoring systems is particularly on the monitoring of heart rate variability (HRV) in canines to identify health and behaviour of the canine [11]. Heart Rate (HR) can monitor the animal's general cardio during physical activities while also functioning as a value that can provide identifiers of stress and immediate health conditions. HRV, which is derived from HR as studies have shown, can be monitored to identify the fluctuations of the Autonomic Nervous System (ANS) and how HRV influences the ANS in regulating emotions through responses of the para-sympathetic and sympathetic system responses to various negative/positive stimuli [3] [8] [12]. HR sensors can be employed in solutions for monitoring pet HR with relative accuracy to that of common, ECG (electrocardiogram) monitors used in pet clinics which is of a commonly agreed gold standard among veterinarians for general health monitoring procedures.

The application of health monitoring systems for measuring HR and HRV in canines is an expensive and invasive process that involves strenuous preparation in order to ensure accurate and efficient health monitoring. A study into the application of less invasive methods of health monitoring in canines [3] referred to the traditional methods of health monitoring in dogs to be invasive due to the application of such systems requiring procedures such as hair clipping, shaving and discomfort when applied to the animal's body for precise monitoring accuracy. Some systems even involve embedding chips into the animal for a more permanent solution of monitoring the state of health of the animal.

As such, the health monitoring system solution should follow the specifications of commercial availability as well as non-intrusive implementation for use outside of clinical scenarios and in the scope of general use for pet owners to monitor the health and behavioural activity of their companion in everyday use.

2.1.4 Geolocation

Geolocation (also known as geotracking, geopositioning) [13] is the method by which objects, people, animals, or otherwise are tracked on the map at a global or localized scale through the use of hardware and software solutions with internet connectivity. A solution for location tracking in a real-time fashion is through device-based geolocation. Device based geolocation is involved in devices embedded with GPS sensor units capable of identifying their location through a Global Navigation Satellite System (GNSS) and identifying their location through a process called triangulation. Due to the ambiguous nature of GPS tracking, the system can be applied to any object to identify its location. There are a number of GNSS that location tracking systems rely on. However, GPS remains the most accurate and ideal choice of GNSS for geolocation tracking. Within the field of animal monitoring, wearable trackers have been designed for commercial use in domesticated animals (such as Tractive [14] and WhistleGO [15]). Furthermore, practical implementation of customized location tracking systems have been largely implemented for livestock monitoring [16] to identify movement and migration patterns as well as the behaviours of the farm animals in open pastures.

Most commercial GPS enabled trackers are also enabled with internet connectivity. GPS devices consist of a GPS receiver that captures the GNSS satellite signals and processes the information into latitude and longitude coordinates. A Global System for Mobile Communications (GSM) module is built into GPS trackers, which also has recently been modernized with General Packet Radio Service (GPRS) to allow them to transmit their location information to internet connected devices and systems like servers or mobile phones with reduced cost, and more established network communication procedures [17]. Due to the cellular connectivity, Subscriber Identification Module (SIM) is included in GPS trackers (not including the SIM card) to allow for provision of a SIM card for cellular service for GSM. More modern solutions for GPS trackers warranted for GPRS as a means by which the trackers can trasnmit their data in a speedier and less costly fashion through GSM/GPRS over a Transmission Control Protocol (TCP)/Internet Protocol (IP) stack. By means of TCP/IP communication, GPS trackers are capable of communicating with resources such as servers and data stores hosted on the internet and send their information to such servers with a low cost footprint, enabling cheaper and real-time location tracking solutions that can be streamed to any application. Figure 2.2 shows an overview of the communication protocols required for a mobile app providing geolocalisation functionalities.

In the application of location tracking of domesticated dogs in training or general pet ownership, the setting is most often urban or sub-urban. As such, the accuracy and application of location tracking and its corresponding technologies should be investigated within such environments, in order to assess accuracy and the advantages/disadvantages that such environments pose for location tracking [18].

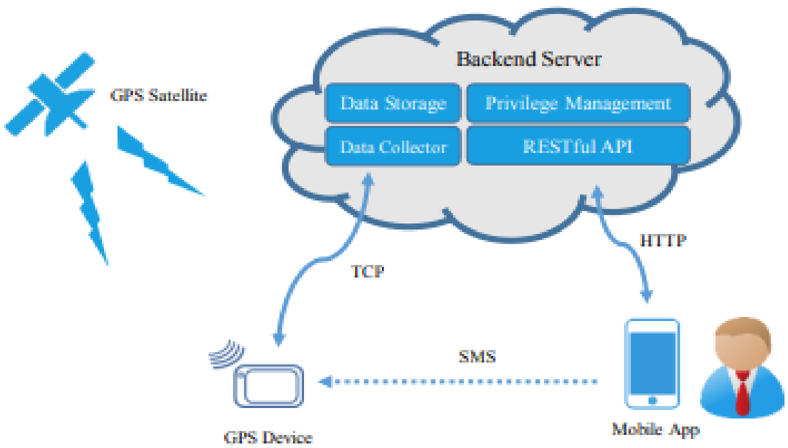


FIGURE 2.2: Example diagram of GPS device integration in an IoT system [2]

2.1.5 Vision Systems

Vision Systems are systems that use imagery from camera devices in order to assess the environment and from it derive and categorize the objects within the image(s) for identifying anomalies in behaviours of objects, or generally identifying certain objects within the vision space. Vision systems have been inspired previously in monitoring of animals and analyzing their behaviours [19]. In the proposed canine monitoring application, the aim was to propose an abstract concept of a vision system in the application to canines to identify potential stressors to the canine with relation to the stress levels derived from HRV. Subsequent evaluation into the applications of such a system were discussed with AADI [20] and considered to be potentially too intrusive within the space that the canine monitoring application is intended to be deployed, where the foster families value their privacy. Furthermore, the resources for vision systems are quite costly with relation to performance, as an efficient read of the canine's surrounding would require a live-feed from a camera which is a series of images streamed at Frames Per Second (FPS). This would prove extremely costly with relation to data gathering, battery life of the IoT device, and communication of data when combined with the requirements of managing health monitoring and location tracking. The live-feed solution would also be the only ideal approach for deriving objects surrounding the canine in a stressful encounter as the alternative is to provide on-demand images when the canine is stressed, which is not always likely to capture the source of the canine's stress. On-demand live-feed could be employed to transmit the surroundings of a canine, although it would likely hamper other sensor transmissions heavily and may impede on the Quality Of Service (QOS) of the overall IoT system, causing periodic anomalies in the monitoring of geolocation and stress levels.

2.2 A Review of Health monitoring system applications to canines

In health monitoring systems, application for monitoring the health of a biological body generally follows similar approaches. In clinical practice, Electrocardiogram (ECG) monitors are most commonly employed to monitor HR and other aspects of the cardiovascular system with relation to some other know variables. The application of such health monitoring systems is similar in humans as it is in animals, the main application difference being at the core of the data-driven systems, the sensors. In human application of HR monitoring systems, it is rather easy to trust that the values of the monitoring system are accurate due to the ease and lack of obstructions between the source of the data and the sensor. In HR monitoring applications within the space of animals, namely dogs, the implementation becomes a bit more challenging.

To address the effective application concerns of HR monitoring systems in canines, a number of validation studies were performed [21] [22] as a practical study on the application of human-oriented Polar™ HR monitors [23] in canine health monitoring. The comparison conditions involved pairing the commercially available monitor against a clinical ECG monitor that is largely considered to be the golden standard for measuring R-R intervals (RRI), from which properties like HR and responses of the ANS branches, namely the para-sympathetic and sympathetic branches for canine behaviour analysis are derived. The importance of the study was to identify if commercially available heart monitors not catered directly to canines would be feasible in the application of accurate health monitoring. Furthermore, the study sought to identify the opportunity in canine health monitoring system applications outside of veterinary clinics with cost-effectiveness and data accuracy in mind, as health monitoring systems are generally extremely expensive for the average individual, and also invasive in that they require uncomfortable procedures for the canine such as hair clipping and application of conductive gel for a good fit, and high accuracy from the heart monitoring sensors. The studies state that the necessity of hair clipping is not essential, although application of conductive gel that improves the accuracy of the polar health monitor was still used. The resulting readings of the Polar monitor were required to be cleaned for a higher level of accuracy due to data anomalies. The final results yielded promising accuracy to the standard of HR monitoring using ECG. [22] Performed the similar accuracy study with ambulation of the canine in mind and saw no major differences except moderate reading discrepancies due to poor fitting of the electrodes of the Polar sensor to the animal which highlights the importance of a fastened application of the sensor to the dog. However, the effectiveness of accuracy has only been proven in limited ambulatory or stationary conditions.

A prototype system involving wireless canine health monitoring [3] suggests a system design that can bypass the invasive procedures, while also highlighting the effects of dog fur on the accuracy impedance of readings of HR sensors due to the fur reflecting the light induced by the sensor. Figure 2.3 represents the prototype system identified the differences in the electrode design and application (with and without conductive gel) in conjunction with ECG electrode readings to identify the differences in readings in terms of accuracy.

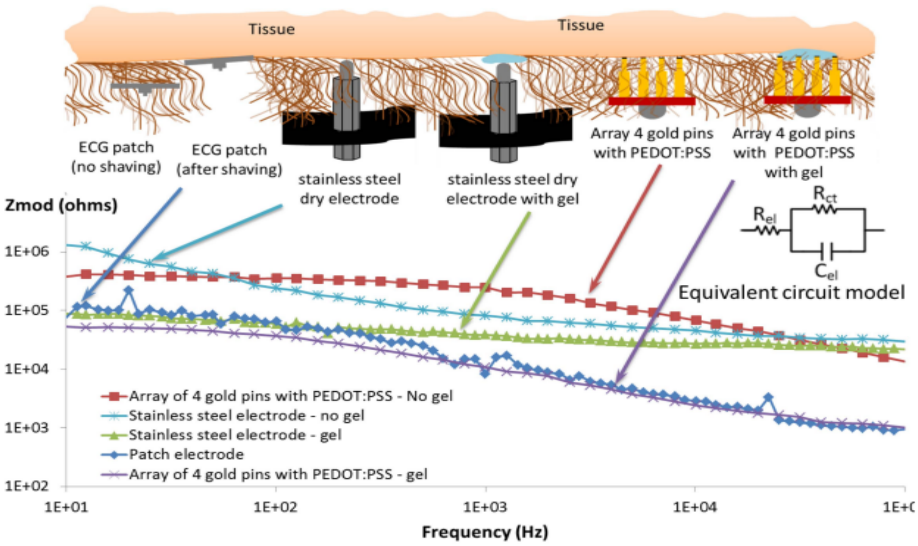


FIGURE 2.3: Accuracy impedance of prototype health monitoring system electrodes with ECG electrode accuracy [3]

Such information is valuable in identifying the margin of error that would come from designing a non-invasive and commercially available health monitoring systems for dogs with basic pulse sensors, granted that they are applied in regions with minimal fur and ideally used in application on canine breeds with a thinner coat of fur.

While a feasible and functional design with relation to health monitoring accuracy is important, it is also imperative to address how HRV is capable of monitoring stress levels in canines. In laymans terms, HRV is the fluctuation in the time interval between heart beats, or RRI. This fluctuation is brought about by two branches of the ANS, namely the para-sympathetic (rest and digest) and sympathetic (fight or flight) branches. As the nervous system operates, each of these branches fight for control. The para-sympathetic branch will reduce the HR while the sympathetic branch will increase it. Thus, a high HRV is an indicator of healthy ANS activity, whereas low HRV is an indicator for high control in one branch of the ANS system, indicating that the canine may be experiencing heightened nervous system response induced by the environment, social encounter or otherwise. A study [7] of HRV in dogs with relation to stress was performed. To identify positive and negative emotional states, the dogs were exercised in a controlled environment with their owners performing various activities to stimulate varied emotional responses for evaluating HRV and its correspondence with stress. The study measured 3 values in order to evaluate stress: Standard deviation of NN intervals (also called RR intervals), (SDNN), Root Mean Square of Successive Differences (RMSSD), and mean RRI, which are required for the calculation of the RMSSD and SDNN measurements.

$$SDNN = \sqrt{\frac{\sum_{i=1}^{N}(RR_i - meanRR)^2}{N-1}} \quad (1)$$

$$RMSSD = \sqrt{\frac{\sum_{i=1}^{N-1}\big((R_{i+2} - R_{i+1}) - (R_{i+1} - R_i)\big)}{N-1}} \quad (2)$$

Figure 2.4: Derived formulas for calculating RMSSD and SDNN from RRI [4]

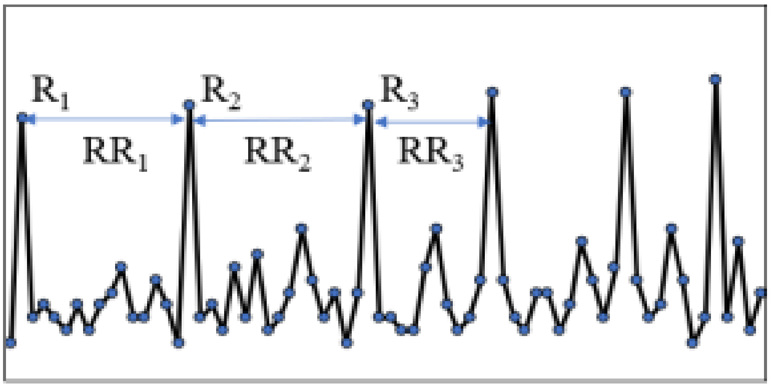


Figure 2.5: Illustration identifying R waves and RRI [4]

Each R wave is representing a beat of the cardiovascular system. The measurement of the RRI appears to be the common standard for the basis in measuring HRV, and consequently, similar clinical and general studies lead to the same approach [10] [8]. Furthermore, both SDNN and RMSSD are mathematical measurements entirely dependent on RRI as they are equated measurements derived from the RR intervals. The paper's summary of results of the investigation into HRV and its relation to the ability of monitoring stress levels of a canine was evaluated. In this evaluation, the study involved exercising the dogs through various activities eliciting specific emotional responses. Before testing, the study collected baseline readings (when canine is not stimulated by any acitvity) of the three variables of SDNN, RMSSD and mean RRI to compare the baseline with the values generated by the positively/negatively stimulating activities. The importance of measuring a baseline reading in monitoring of HRV is an imperative task before data collection and analysis can be made as HRV is a highly characterized value that is determined by the temperament, age, and overall state of health [24] [25], as such there is no obvious measure by which HRV can be universally evaluated. In comparing the baseline with the values collected from the activities, the study found only decrease in SDNN in positive engagements, whereas in contrast, RMSSD was the only value to decrease in negative situations. Comparatively, a similar study [10] was performed where dogs were exercised through positive/negative social and reward encounters. In this study, SDNN was not collected and in its place, high frequency (HF), low frequency (LF), and HF:LF ratio was collected. In this study, the resulting conclusion identified a lower RMSSD in conjunction with a lower HF to identify a positive emotion. This is a direct opposite to the initial study suggesting RMSSD to decrease in negative scenarios, although it was highlighted that *only* RMSSD decreases. Notably, both of these studies were performed with relation to practical findings through controlled environments, and not simply hypothesis. This suggests multiple methods by which stress levels of canines can be monitored. This diversity in monitoring HRV is further suggested in an in-depth study of time-domain and frequency-domain measurements for monitoring HRV [26].

A paper documenting the implementation of a novel stress monitoring system using HRV was performed [4]. This paper documented the results of using SDNN and RMSSD in monitoring HRV fluctuations and its correspondence to stress and found promising evidence of usability and reliability through examination on the system readings with relation to stimuli on the dog. The system is also subsequently non-invasive by design, incorporating basic pulse sensors for measuring the HR of the canine, and not necessitating hair clipping or application of gel. The proposed system suggested that application of the sensor be placed on regions with minimal/no hair, which poses challenges for breeds with heavy fur coats. The system design allows the IoT device to strap on to a harness for walking a dog. Since the harness is in direct contact with the dog, the sensors are able to maintain skin contact for persisted data collection without much data loss.

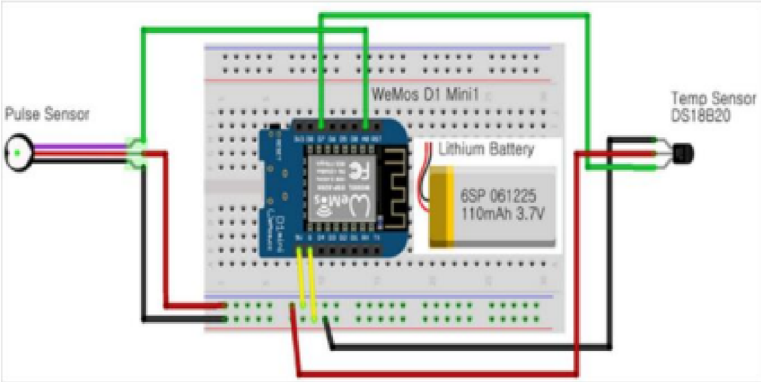


FIGURE 2.6: Image illustrating architecture of health monitoring device [4]

Figure 2.6 presents a WeMos Arduino mini [27] with built in WiFi capabilities for remote data transmission. The system utilized Thingsboard [28] as a cloud data store. For measuring the actual stress levels of the canine, the system employed baseline reading functionality, where readings were estimated to be 15 minutes of measuring the RRI of the HR of the canine, provided by the pulse sensor default inter-beat intervals (IBI) reading capabilities, where IBI is just another term for RRI. The baseline readings were documented to be necessitated for the similar reasons, so as to know what the standard HRV is of the canine in a neutral psychological state. Furthermore, the baseline readings are what the subsequent, real-time HRV reading calculations for SDNN and RMSSD are paired up against in order to identify fluctuations in either measurement to determine the mental conditions of the canine (measured in the system as "Good" "Not Good" and "So So"). The system does not take into account however the evident fluctuation in HRV due to posture [29], and as such may be prone to some challenges in monitoring of stress levels in situations where the dog is subject to constant change in posture, particularly during play-time due to lying down, sitting, rolling around. Generally, HR is subject to drastic changes in intense exercise and dogs are typically known to exhaust themselves during play-time. This in turn makes such health monitoring systems less functional in such scenarios, and are better allocated to walking or more controlled activities.

The analytical aspects of the system (such as the baseline readings, evaluating SDNN and RMSSD from RR intervals, and determining stress levels) were handled by the Thingsboard cloud system as the operations involved in calculating SDNN and RMSSD and subsequently evaluating stress is an intensive procedure, better handled by an external process rather than evaluated by the raw resources of the mobile device which would impede the responsiveness of the application. In measuring the stress levels, the system employed a technique using the Wilcoxon rank test [30] to determine the levels of change in RMSSD and SDNN with relation to the initial baseline readings taken to determine the stress levels. A higher SDNN meant less stress while a lower SDNN indicated heightened activity, indicating higher stress. The RMSSD was used as a measure for the variability in the HR over a short period of time. The Wilcoxon rank test involved pairing the baseline readings against the real-time data derived from the SDNN and RMSSD measurements. Each comparison of baseline and data granted a minus or plus rank which identified the decrease/increase change in SDNN and RMSSD, through which the assumptions of "Good", "Not Good", and "So, So" were made. This operation, in conjunction with the wilcoxon signed rank test, sums a set of baseline and current readings of either RMSSD and SDNN and sums the minus and plus differences between baseline and data to get the final value after combining the minus and plus ranks together against a critical value that determines the condition of a significant change in either RMSSD and/or SDNN. Based on this information, the three different condition assumptions are made.

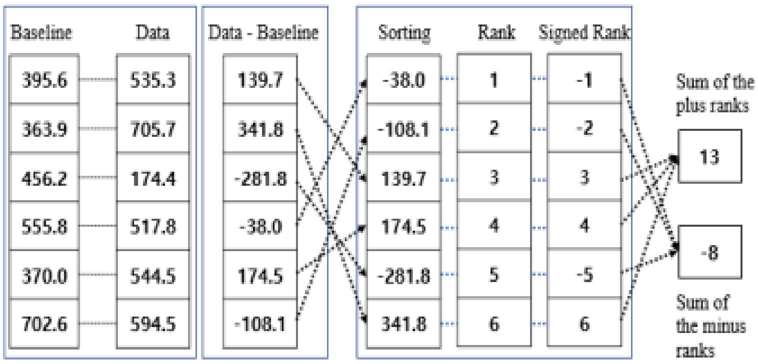


FIGURE 2.7: Wilcoxon rank test process for measuring baseline against data, and the output [4]

TABLE 1
EXPERIMENTAL RESULTS FOR EVALUATING STRESS ACCORDING TO ACTIONS

Actions	*SDNN*	*RMSSD*	*Results*
Petting a dog	*8*	*10*	*So So*
Telling a dog to go for a walk	*-9*	*6*	*Good*
Giving a dog snack	*-43*	*9*	*Good*
Playing with a toy	*5*	*5*	*So So*
Going around alone	*3*	*9*	*So So*
Sitting down	*8*	*9*	*So So*

FIGURE 2.8: Resulting SDNN and RMSSD values with relation to stress message output [4]

2.3 A Review of GPS Tracking Systems Applications

A low-cost real-time tracking system for monitoring the location of elderly individuals was developed [2], and its accuracy was assessed within an urban environment. Although the study does not pertain conceptually to tracking canines, it still highlights the general approach process in implementing location tracking with IoT device and mobile applications. The system possesses an IoT stack consisting of the tracking device embedded with GPS/GSM/GPRS, a backend server for storing location data and validating data requests through a REST application programming interface (API), the mobile application that displays the location data received from the server, and a TCP/IP gateway for communications. The tracking device communicates the location data at a specified interval over TCP/IP. The structure of the location data over TCP/IP communications is identified as American Standard Code for Information Interchange (ASCII) and dissected.

TABLE I. STRUCTURE OF A LOCATION MESSAGE

Field	Description
1	Message type code
872501000030793	IMEI
1463468426	Unix timestamp
113.563367E	Longitude
22.329371N	Latitude
060	Battery status
0460	MCC
0000	MNC
2623	LAC
0e09	CID
98	RSSI

FIGURE 2.9: Dissected location message structure and the associated values [2]

For ensuring low-cost measures in location data transmission, the system opted for GPRS based TCP/IP communications for their reduced cost and increased speed over transmitting over GSM SMS based communication, as the investigation identified prolonged periods of SMS messages to prove costly. The IMEI portion of the message is a unique identifier (UID) for the tracking device that is essential for identifying the origin of the location messages with the IMEI being the value that identifies the device that issued the information. The paper notes the occasional loss of reception from GPS and have derived an auxiliary location estimation method. When GPS experiences loss in reception, the longitude and latitude data returns as missing data from the tracking device. The system's auxiliary method for estimating location consists of using beacon-based tracking (otherwise known as Cellular Network Positioning (CNP)), although it does have reduced accuracy as compared to GPS. A referenced paper [31] highlights this lack of accuracy in CNP, stating that the accuracy can deviate into the kilometre range, which means it is extremely inaccurate and may cause more harm than good in the canine monitoring application when attempting to identify the correct location associated with stress in a canine based on stress data. However, an auxiliary location measurement would yet prove useful, especially in the canine monitoring application as the walking environments can present highly obscured locales for GPS based tracking, such as forests with a dense canopy or obstructed urban areas with high buildings.

FIGURE 2.10: Historical location tracking indicating jagged tracking, possibly caused by auxiliary CNP tracking [2]

For displaying the location data broadcast from the tracker to the server, a mobile application retrieves the location data through a REST interface by communicating with the server, and displays the location data using a map service (China's own equivalent to Google Maps [32], Amap [33]) that handles the presentation of the location data on a graphic map of the location. The implementation notes that GPRS becomes unavailable on occasion due to the absence of an internet connection. In such cases, the system notes that using the auxiliary SMS capabilities of GSM could be used to persist communication of GPS data as the culprit of periodically absent location tracking may also be the lack of internet services, rather than GPS signal loss.

With relation to the hardware for the canine monitoring application, an in-depth paper [34] documenting the implementation approach of an efficient GPS tracking device was performed. The system was noted to be reusable for animal monitoring and goes into great detail highlighting the hardware solution of the system as well as the over-arching libraries and APIs used. With regard to GPS sensors, the study performed analysis of sensors with relation to performance, low cost, and efficiency with relation to low power consumption. They addressed the GPS Neo6M sensor module to be the most sustainable, not compromising expected performance due to design aimed at reducing power consumption. Furthermore, the study identifies the required networking module GSM Sim, particularly the GSM Sim800L simply as a result of the low-cost. This device is necessary to allow for the system to broadcast and receive data from the sensor hooked up to a micro controller, using a SIM card to be eligible for cellular service and connect to networks. Finally, the paper idealized the Arduino Uno microcontrollers [35] in their approach due to the popularity of the micro controller as well as its user-friendly interface, making it the ideal approach for beginner developers or start-up projects.

(a) (b) (c)

FIGURE 2.11: (a) Arduino Uno micro controller (b) GPS Neo6m Sensor (c) GSM Sim800L Module

Arduino is particularly powerful due to being open-source, allowing experienced developers to create useful libraries and utilities that can be injected into any project. The paper notes the potential read anomalies that may occur in Arduino micro controllers in the case of multiple sensors transmitting data. Additionally, the paper notes the importance of ensuring QOS when adding new sensors with relation to communication of data, and processing as a poor implementation can lead to overall IoT system performance loss due to IoT systems being inherently data-driven. Similarly, for the cloud approach, the implementation called for Thingsboard as it is the ideal cloud solution for IoT based systems for gathering, and processing the information for presentation in mobile applications. For gathering the data, the implementation used a TinyGPS [36] library that handled all the data gathering operations, and came pre-packed with a number of functions for selectively collecting information from the GPS sensor. To display the information, the approach involved a Google Maps API [37] that translated the longitude and latitude information into a visual presentation on a rendered map of the surrounding area.

Chapter 3

Problem - An IoT, Sensor-Based, Non-Invasive Dog Training Programme Assistant)

3.1 Problem Definition

Becoming a dog owner is never an easy feat due to the number of demands and complications that arise from owning a canine. From nurturing them and training them when they are young to behave correctly among humans and other animals alike, to making sure they are kept healthy and safe. A high level of knowledge and often times sheer intuition is required to do this correctly.

Unfortunately, many dog owners have not had any formal experience or training in owning a dog or better yet, understanding their true nature through behaviours and psychological study as a certified canine trainer would. As such, many owners need guidance in understanding their canine to improve and enhance their general knowledge of how a dog behaves, which in turn can help them interact and exercise their dog at a higher level of understanding. This can attribute to better relationships between the dog and the owner, and subsequently to better ability at training the canine through a heightened understanding of their temperament and psychology. This heightened level of understanding can subsequently contribute to fostering an informed canine training environment for evaluating their temperament and suitability for specialized roles.

Canines are extremely characterized, with a lot of their behaviours not directly correlated to some known psychological patterns or parameters as a result of taking from domestication [38], or the general environment they grew up in. Having a means by which an individual can identify what induces stress in a canine, they can rule out certain locations or events from the canine's life to make for more pleasant experiences for the canine, or in other cases, make induction of canines or training them more streamline when the psychology of the canine is elaborately understood and can be monitored.

3.2 Objectives

The core components sought after to achieve the primary objectives within the problem space are comprised of the following: Location tracking system, health monitoring system, mobile application, cloud storage/processing, and internet connectivity. Each of these key concepts will attribute to answering the problem of improving interactivity with our canine by means of IoT.

The objectives of the problem are to develop a mobile application that can improve the interactivity with our canine by providing meaningful, and visual representations of how the canine is feeling, how their well-being is through the monitoring of their heart rate and deriving HRV for interpreting stress levels, and informing the caretaker of the location of the canine during bouts of stress to potentially identify stressful locations for the canine in conjunction with the evaluated stress data. Additionally, the stress evaluation can be used in conjunction with training environments (particularly within induction of new guide dogs during foster family training) to evaluate suitability of the canine by examining stress levels during training exercises.

Due to the setting of the application of the monitoring system, the system requires a level of finesse with regards to its functional design as the implementation of the physical system must be applicable to a harness that is fastened on the canine, as the canine is the source of data for the monitoring system, and the system requires full, and secured contact with the canine at all times for persisted data collection.

3.3 Functional Requirements

The functional requirements summarize the core components and concepts required as part of the overall functionality with regards to overall design and proposed architecture, implementation, and operation.

3.3.1 Location Tracking Unit

The mobile application requires location information to identify stressful locations for the canine with relation to stress data derived from HRV. As such, a location tracking unit will be required, that is capable of determining the location of the canine on the map in order to track it, so that this information can be captured.

3.3.2 Heart Rate Sensor

In order to determine the health and stress levels in a canine, the system will need to have some sensor device by which it could perform this. A heart rate sensor is capable of capturing the HRV of the canine which, in research and science, is a largely employed method by which emotional states in animals can be deduced. The selection of heart rate sensor should be one of no invasive procedure requirements as the intent is to deploy this system outside of clinical scenarios and more so in general engagements, and or training of canines.

3.3.3 Network Connectivity

All of the above systems involved in the project demand that there be network connectivity in order to have a means by which the information from these sensors can be transmitted to a destination (e.g. the mobile application) and vice versa. While the data is collected by the sensors, it requires to be output to a destination that can process the information in order for the mobile application to display it in the intended way. As such, network connectivity is required for data transmission.

3.3.4 Mobile Application

The system will be developed within a mobile application due to the premise of the system involving a high level of mobility as a result of the general mobility of the dog. The core of the operations will be a mobile application from which the caretaker of the canine can view the information about the dog's health, mood, location, etc.

3.3.5 Cloud Services

The data returned from the IoT device sensors will largely be unfiltered and unchanged. Particularly for the health monitoring, the HRV data needs to be derived using mathematical formulas in order to extract the necessary values from the heart rate data that pertain to the assessment of mood in canines. These operations may be too intensive if applied to the IoT device and would hinder the operational capabilities of the IoT device to transmit the data, diminishing the overall operational speeds of the overarching system as data will be transmitted too slowly. Similarly, these operations should be avoided by the mobile application as this may drastically reduce the responsiveness of the application with relation to delivery of information to the end-user in real-time which would defeat the purpose of the real-time monitoring system specification.

3.4 Non-Functional Requirements

Non-functional requirements encompass the more "niche" and perhaps stylistic aspects of the system that are not tied down by pure requirement for the system to function. In the scope of this project, the system's non-functional requirements revolve around the way the data could be represented visually within the application and quality of life improvements for the system to prove more convenient and secure for the user.

3.4.1 Calibration

While the system can monitor HRV without calibration, the accuracy will likely be much lower. In monitoring HRV, the value of HRV is extremely volatile and highly characterized by each passing day in canines. As such, HRV measurement should be taken for a number of minutes as a means of calibrating the IoT system for the day's HRV measurements with relation the the HRV data collected before operation. This would increase the accuracy of the stress measurements.

3.4.2 Visualisation of Data

The visualisation of the data is an important aspect of the non-functional requirements. The application part of the system will need to represent the information regarding the canine's mood and health in some stylized fashion. Ideally, the application will represent the data with some visual imagery of the canine in question (such as an icon of the canine) that is consistent with some additional information about the, canine, like the family they belong to, and the breed.

3.4.3 Battery Life Notification

The system could inform the user of battery life of the sensors that are involved in populating the mobile application with important information about the canine, such as their location, physical health, and mood. The form of alert can be indicative to the owner as to why the system may suddenly be miscalculating, as battery life is often a candidate in reduced performance of electrical components. Since conceptually, the system is IoT based, its reliance on the persisted function of the sensor systems that provide it the data cannot go unnoticed. As such, their performance would ideally be monitored to ensure a high level of functional security for the IoT ecosystem of the system's architecture.

3.4.4 Connectivity Notifications

Loss of connection during operation can be induced by poor network coverage or general packet loss, which the system could account for with notifications to inform the user of the shortcomings of the system. This could help the user to realize the cause of the malfunction in monitoring or better yet understand the conditions the system requires to be in to maintain operation (certain physical areas could be avoided if they have poor network coverage).

Chapter 4

Implementation Approach

This section goes over the architecture of the system and the implementation approach in terms of development, but also the process of progression through implementation. Moreover, this section covers in detail the hardware and software technologies likely to be involved in the implementation of the project as a result of the technical requirements of the system. It covers the various system infrastructures, from IoT device hardware, cloud solution, and the mobile application for displaying the data. The reasoning behind the technologies and hardware are detailed with relation to the to-be features of the canine monitoring application. The methodical approach to solving the more technical problems is also covered within the context of what is required of the system. Evaluation is also assessed to identify the stages of development of the system as well as some prototyping to give an envisioned high-level view of the system.

4.1 Architecture

The IoT based canine monitoring application will comprise of the IoT device housing and managing the sensors, a cloud solution for processing and persisting data as necessary, and finally the mobile application that communicates with the cloud solution for displaying the cleaned data in an orderly manner for the end-user to visualize in a meaningful way. The IoT device make-up will largely encompass the hardware side. The technologies will encompass the framework(s) required to build the mobile application, the API's involved in connecting the cloud solution with the mobile application, and powering the processing elements of the app for managing raw data, and the IoT device operations.

4.1.1 IoT Device

The IoT device consists of a number of parts in order to operate. Two sensor devices are required to collect the data required, which are the heart rate and the location data from GPS satellite pertaining to the canine. To collect the data outputs by these sensors, a central computing system is required to connect all of these parts together and will involve a microcontroller to act as the computation device. This data requires to be transmitted remotely, and as such the device will also require a module capable of providing WiFi capability for remote communication. Finally, a power source is required to power the sensor units and the microcontroller, which can be achieved using batteries with battery holder modules and a sufficient power output for the various components of the IoT device.

4.1.2 Microcontroller

The microcontroller of choice for this IoT device will be an Arduino Uno for its reduced complexity and friendliness with beginner developers. Furthermore, its open-source capabilities allow for quick injection of complex operations through sourcing libraries developed by professional developers of the Arduino community. Arduino is also supported by most if not all sensor systems and modules, making it less challenging to source sensor components and other modules for the IoT device with relation to compatibility. There are many variations of Arduino Uno that may include built-in features, though the IoT device we require does not call for any particular configuration of Arduino Uno and any model should suffice. The motive in selecting the appropriate piece of microcontroller hardware will be down to cost, reliability and size as we want to ensure that the resulting IoT device does not possess dimensions that will make it challenging to mount it onto a harness.

4.1.2.1 Breadboard

A breadboard is a basic construction base for building systems involving microcontrollers, sensors, and other modules. The reasoning behind the necessity of the breadboard in this project is so that all the connectivity between the modules is possible. Notably, the GSM modules typically connect to breadboards, and source their pins from the breadboard which are connected to the microcontroller for power. Additionally, some modules may require similar Voltage (5V or 3.3V that Arduino microcontrollers provide) and the Arduino boards only possess one of each pin. The breadboard can extend the number of pins for each desired voltage, allowing the IoT device to power multiple modules with identical voltage requirements.

4.1.2.2 Pulse Sensor

For deriving HRV and stress levels, a measurement of the heart rate is first required. To do this, a heart rate sensor called "Pulse Sensor" [39] will be used, which is commercially available, low in cost, and reliable in operational speed. The Pulse sensor is fully compatible with Arduino and should fit the role of the heart rate sensor well. Furthermore, the pulse sensor is capable of monitoring valuable data such as IBI which is the equivalent of the R-R interval, just with a different name. This IBI is required for the processing stage of the HRV and stress derivation operations and the pulse sensor should fit the role of the heart rate sensor for the purposes of this IoT device in monitoring canine stress levels.

4.1.2.3 Location Sensor

When measuring the stress levels of the canine, the system intends to log the location that the canine is currently in during the bouts of stress in an attempt to identify potentially stressful locations. For tracking the location of the canine, a GPS Neo6m sensor would be ideal within the project due to its full compatibility with Arduino, low cost, and reliability. Furthermore, this sensor is the most documented and widely used of all GPS sensors, proving its reliability and ease of development. The device returns longitudinal and latitudinal data which can be translated into coordinates on a map.

4.1.2.4 GSM module

While the microcontroller can collect the location data and heart rate data of the canine, this information is required on the mobile application after having been processed in a more meaningful and readable fashion, and the microcontroller has no means by which it can do this on its own. A GSM module with GPRS capability can be used to achieve remote network connectivity. There are many derivations of the GSM modules but the GSM Sim800L is the most popular and widely available GSM module. This module provides network connectivity through cellular connection by the use of GSM/GPRS which requires a SIM card. This process will result in added cost of the SIM card, but persistent network connectivity can be achieved this way, allowing the IoT device to issue the raw sensor data remotely to an external source.

4.1.2.5 Battery Module

A battery holder module is a basic component that is used for the sole purpose of housing a battery or multiple batteries for powering IoT devices and their various components. These battery holders can be slotted with batteries and provide a breadboard with a voltage that can supply necessary power to the modules connected to it. In our case, the devices powered would be the Arduino and the connected pulse sensor and GPS sensor, and the GSM module that would connected to the breadboard.

4.1.3 Technologies

The canine monitoring application requires a number of additional layers for gathering, processing, and displaying the data to the end user. The canine monitoring application will ideally deploy a cloud based solution for processing and managing the raw data output from the broadcast, from the sensors. This is the ideal approach as the processing of the IBI data from the pulse sensor into HRV and then deriving the stress levels of a canine may likely be an intensive procedure, and the process of calculating the values required requires a means by which IBI data can also be stored. A cloud solution like Thingsboard can handle the telemetry data from the sensors and store them in database instances, while also providing data processing capability. The idea in this approach is to have a means by which the sensor data can be persisted, and processed in a way that does not impede the overall performance of the mobile application interface that the user will inevitably interact with to view the information about the canine in a real-time fashion. Furthermore, API's within the mobile application will be deployed for translating information such as location and stress in more visually sensible ways.

4.1.3.1 Thingsboard Cassandra

Thingsboard uses Cassandra database instances for storing data. As Thingsboard is a server-side solution for IoT projects and systems, the database is configured to especially suit the requirements presented by the persistence of sensor data. The persisting of sensor data in the canine monitoring application will be required for deriving historical location tracking with relation to bouts of stress in the canine. Similarly, the derivation of stress from HRV data requires RMSSD and SDNN calculations which require multiple IBI readings to output any meaningful information.

4.1.3.2 Thingsboard Rule Engine

The Thingsboard Rule Engine is a "complex event processing system" built into Thingsboard that allows you to transform incoming telemetry into whatever way is required using mathematical formulas or general logical processes. This Rule Engine is intended to be used in the canine monitoring application for cleaning up incoming telemetry from the GPS sensor as required (if at all required), as well as implementing historical tracking, but most of all to transform the incoming IBI data into meaningful HRV values (RMSSD and SDNN) and stress level values through mathematical equations. In addition to this as per study of using HRV to monitor stress, a standing HRV is required to be captured so as to have a basis against which subsequent HRV values can be mapped against to derive stress level in the canine. The Thingsboard Rule Engine can be configured to have multiple Rule Chains (effectively a sequence of operations), and one of these Rule Chains can be engaged to process the incoming IBI into labelled HRV data so that the subsequent operations know what to map new incoming HRV data against to derive stress levels.

4.1.3.3 React Native

React Native [40] is the go-to software framework for developing the mobile application as it is a good fit for the canine monitoring application for its robustness in features, and ease of development and deployment. React Native is fully open-source with the React community being extremely active in developing ready-to-use libraries that can be employed within the mobile application to inject it with robust operational capabilities that work right out of the box. This level of accessibility to powerful libraries will aid in expediting the development phase of the application and aid in whatever challenging operations may be necessary for the application to deploy as much of the implementation phase will likely fall upon the design of the IoT device and the processes for deriving stress data. Similarly, React Native supports many API's that can provide visually pleasing graphics as well as powerful operational capabilities for handling the location data and stress derivation data from Thingsboard within the application.

React Native maps is a powerful library for developing mobile apps that involve some form of map, location, or location tracking. This library is essentially google maps but configurable by developers for implementing historical location tracking, and generally performing location tracking operations. This will be the go-to map API for handling the location tracking features as well as other metrics such as distance travelled when identifying reasonable activity conditions.

4.2 Risk Assessment

4.2.1 Usage conditions - Mild

The canine monitoring application will likely be best suited for limited but not entire lack of ambulatory conditions, enough to accommodate for the typical training and general activities performed with a dog and owner/caretaker. When applied in more rapid and mobile conditions, severe sensor data loss could occur, effectively minimizing canine telemetry readings. This is as a result of the possible challenges with maintaining reasonable integrity of the IoT device when mounted on the canine with relation to managing appropriate accuracy of readings from primarily the heart rate sensors. The application of the system can be work shopped to be better fastened to the canine but the intent is to ensure that the canine has minimal/no comfort loss so long as the device is attached to the harness of the dog.

4.2.2 Poor network connectivity - Moderate

During the operation of the canine monitoring application, the gateways and networks that interconnect the IoT device data output with Thingsboard and finally the mobile application are what keep the system active and outputting real-time updates. Sometimes network availability might falter and cellular might have some hiccups. In such cases, it could be damaging for the real-time capability of the system as data may be lost during operations, causing momentary radio-silence from the output of the sensors. To mitigate the issue, active notifications from Thingsboard (using Device State Service) should be issued to the user from the app to inform them of momentary communication loss in order to ensure that the user knows the conditions within which the system may not be as operational (e.g. if they enter regions with poor cellular).

4.2.3 GPS reception - Moderate

GPS reception can stagnate at times due to the region within which location is required. GPS does not perform flawlessly when met with conditions involving high buildings or heavily obscured areas like forests with thick canopies. In such cases, the monitoring application may not have a means by which it can reliably identify the current location of the canine when accounting for a stressful scenario for the canine.

4.2.4 Application of Pulse Sensor - Critical

When considering the application of the Pulse Sensor to the canine, the main concept to consider is that the Pulse Sensor is not directly designed to accommodate for the unique biological makeup of a canine. The hair, thicker/fattier skin both influence the effectiveness of the Pulse Sensor and as such, care must be taken in applying the sensor to a region of the canine that has minimal hair and layers of skin for the best chance at retrieving reliable heart rate readings.

4.3 Methodology

This section describes the methodology to be followed during the implementation phase, primarily within the confines of the more complex operations involved in achieving the envisioned canine monitoring application.

4.3.1 Stress data derivation from HRV

For finding out stress from HRV data from heart rate IBI/R-R intervals, the formulaic approach proposed by [7], and the implementation approach proposed in [4] will be followed. The reason behind this decision is due to the potential complexity in deriving stress from HRV, with the approach in [4] seeming to be the most secure while also providing sufficient scientific evidence. Further practical results from [4] yields promising results from implementing the formulaic approach for identifying stress from HRV. The only thing left to consider from this approach in the implementation of the monitoring application, is to design a procedure that can determine when recording of location data should commence given some level of stress with relation to time, as the derived stress data may well fluctuate between the thresholds for determining the levels of stress the canine feels.

4.3.2 Location tracking approach

In the canine monitoring application, the motive behind the location tracking is to identify the location of the canine as a response to a significant increase in stress. As a cause of this approach, the idea would be to intertwine the output of the stress data with the output of the geolocation in order to trigger location tracking during bouts of increased stress.

When the derived stress data is evidence of significant distress in the canine, a function would be fired that would begin logging the location coordinates during the stress time-frame, while logging the length of the stress bout in order to identify time of recovery from stress. These location coordinates would then be stitched together using a separate function to create a trail of location, much like historical location tracking. Each of these historical location tracking events would ideally be stored in a database instance on Thingsboard and marked with an identifier, so when the data requires to be retrieved for examination by the app, the app can retrieve all the coordinate details of that stress event to map out the location of that stress event on a map API on the mobile application. This functionality can be extended in a historical activity view where the user can view activity records and the stressful locations mapped during said activity by retrieving them from the cloud.

4.4 Implementation Plan Schedule

The project implementation plan is separated out into phases, each phase determines the consequent approach as a result of the previous phase to solving the problems of the new phase.

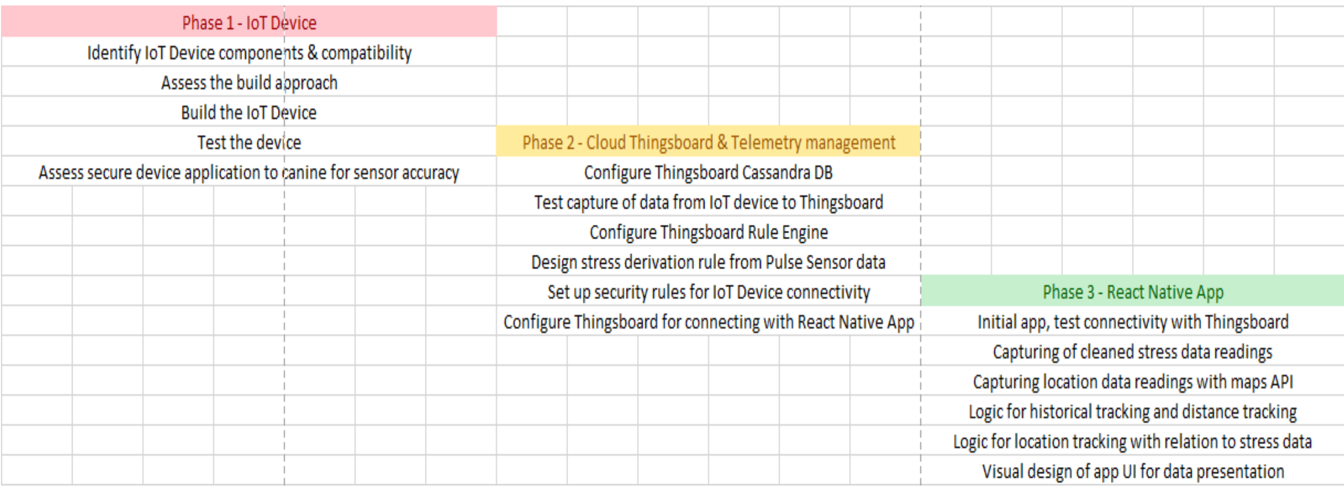


Figure 4.1: Excel graphic illustrating progress chart in phases

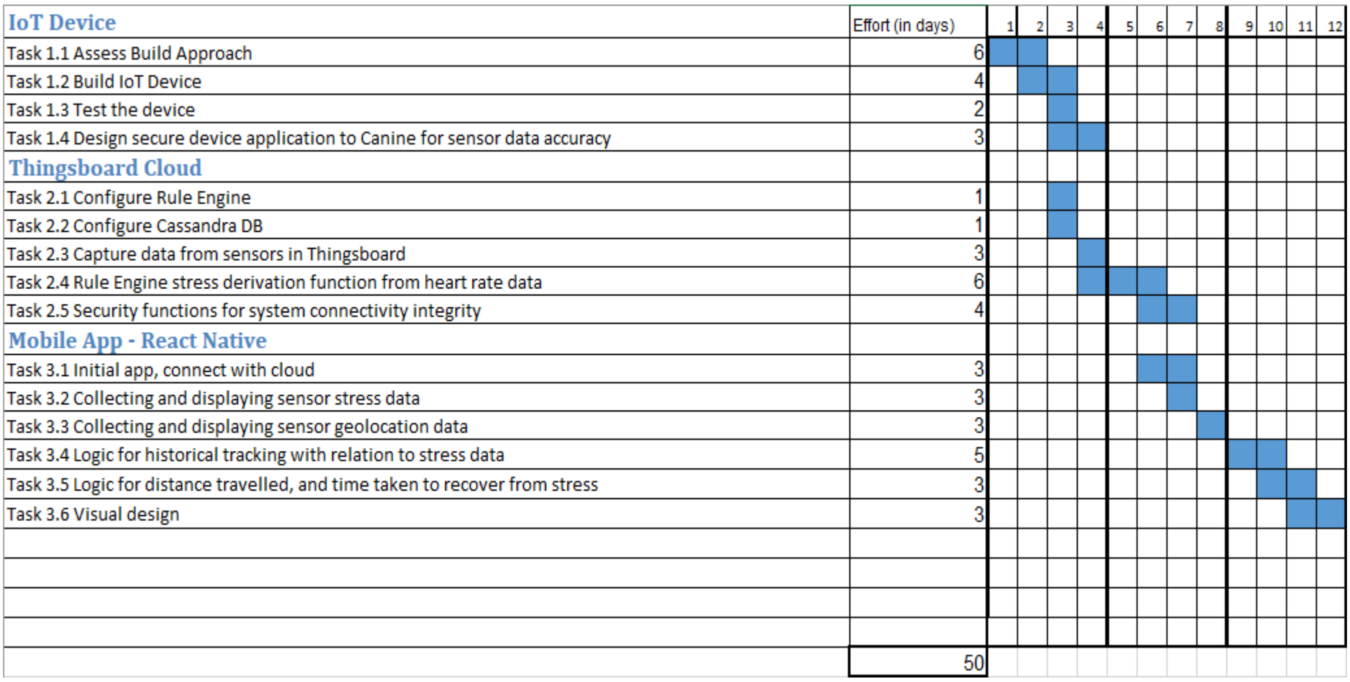


FIGURE 4.2: Excel Gannt graphic illustrating estimated implementation phase progression. Highlighted blocks indicating time working on task during week (weeks 1-12).

4.5 Evaluation

The aspects of the system functionality below are good derivatives for evaluating the level of progress in achieving the envisioned canine monitoring application. These functionalities are good identifiers to how much of the desired system has been completed, and how much of the system is left to be finished.

- IoT device has internet connectivity and can communicate remotely.
- IoT device can issue location and heart rate data to cloud.
- Cloud can process the heart rate data into stress.
- Sensor data is gathered, processed and updated on app in real-time fashion.
- System has calibration for gathering baseline HRV for subsequent stress derivation.
- App can display stress data derived from heart rate.
- App can track location during stress and issue the data to be stored in the cloud.
- Activity reporting from app with location data during activity for persisting in cloud.
- Activity report view in app to historically view stressful locations during said historical activity.

- App informs user of connection loss with Cloud or Cloud with IoT device.

4.6 Prototype

This section seeks to illustrate a basic but concise design of the system that the canine monitoring application will employ. The general functions and components are highlighted in this prototype.

Figure 4.3 illustrates the proposed prototype of the to-be system and how the various components (IoT device, cloud, mobile application) are interfaced and connected. The illustration also documents the possible communication across the gateways between these components that combined together create the complete system for the canine monitoring application. The prototype diagram here is a basic visualization without making any additional assumptions as to how the final system may function, though it covers the key aspects of the generic concept and what is expected overall.

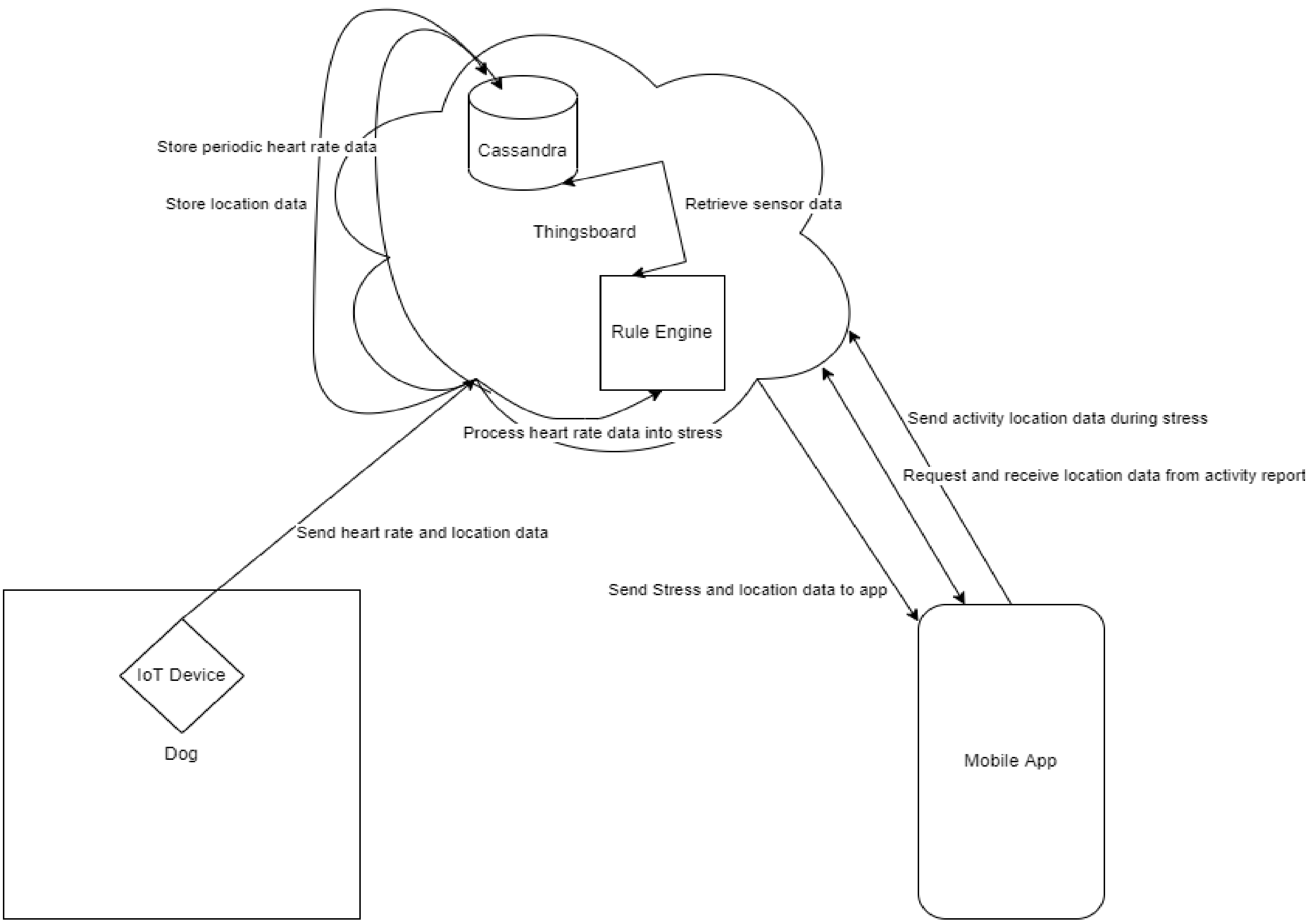


FIGURE 4.3: Basic, high level system diagram of the canine monitoring application.

Chapter 5

Implementation

The implementation of the proposed IoT dog monitoring system revolves around the evaluation of a dog's stress levels, based on their HRV. Fluctuations in a canine's heart rate can be monitored and analysed in conjunction with their activities in order to identify potential stressors for the dog. This form of monitoring can be leveraged further with visualization systems, to provide additional insights into a canine's overall temperament and nature with relation to their regular activities.

A monitoring system such as this may provide much needed insights to dog owners who are wishing to enhance their dog-owner relationships, but particularly for guide dog trainers, who are looking to enrich their training and enlistment procedures through an enlightened understanding of the dogs they may work with.

5.1 System Architecture

The architecture of the dog monitoring system involves a layered end-to-end IoT system architecture consisting of a data source comprised of an IoT microcontroller module with a Pulse Sensor for gathering ECG data, a socket server processor for collecting the ECG data broadcast by the module from the Pulse Sensor via socket connection, a front-end web app visualization client for visualizing the derived stress data from the processor server, and finally a back-end for bridging the communication gap between the front-end and the socket server for the web app to access the processed information.

5.1.1 Pycom Microcontroller

For the IoT aspect of the system, a Pycom FiPy ESP32 microcontroller was used for the purposes of this dog monitoring system. The microcontroller is equipped with Lorawan and Wi-Fi connectivity capability which enables remote broadcast of data. This microcontroller can be used to process the necessary metrics output by the Pulse Sensor to an external source like a processing server, which can perform necessary stress derivations.

The Pycom device is based in MicroPython and as such does not possess a vast library of interfaces that can be immediately applied to various sensor specifications, and as such, requires to be manually coded.

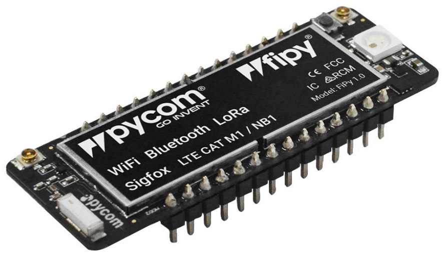


FIGURE 5.1: Image depicting the Pycom FiPy ESP32 microcontroller

5.1.1.1 Expansion Board

An expansion board can be configured with Pycom microcontrollers as a means of making it easier to interface and connect modules to the microcontroller. The expansion board provides a simple, beginner friendly accessibility to the microcontroller pins in order to simplify the process of connecting sensor modules to the board.

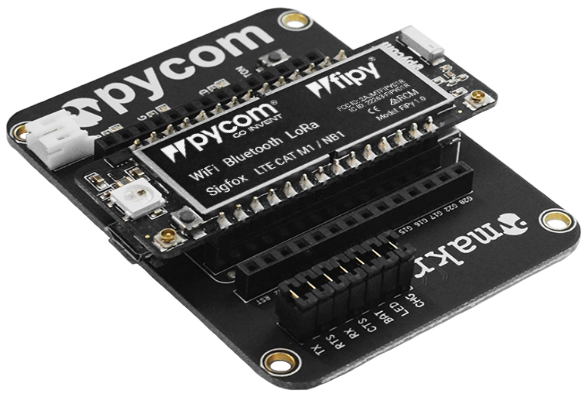


FIGURE 5.2: Image depicting the Expansion Board and FiPy ESP32 module fitted on top

5.1.1.2 Pulse Sensor

The same Pulse Sensor as initially proposed was used in the development process of the dog monitoring system. The Pulse Sensor was inspired by Arduino based development, and as such, only provided pre-written libraries with the necessary functionality to operate the Pulse Sensor for Arduino based projects. As the overarching microcontroller used in the implementation was a MiroPython based one, the code solution for evaluating a discernible ECG, and subsequent collection of metrics necessary for evaluating stress based on HRV had to be written manually.

The performance of the Pulse Sensor in ambulatory conditions was too unstable when attempting to collect sufficiently accurate readings. This is due in part to the sensor responding very strongly to slight pressure changes between its sensor and the surface it is monitoring for ECG signals. Due to this hardware limitation, the ECG monitoring was strictly limited to stationary conditions. Subsequently, the overarching system was evaluated locally due to the lack of ambulation, no mobile power source was required to power the microcontroller module and the Pulse Sensor (i.e. batteries). Similarly, as a result of inaccuracy during ambulation, the location sensor was also not included due to absence of mobility.

Figure 5.3: Image depicting the Pulse Sensor module

5.1.2 Python Processor Server

A processor server based on python was implemented to which the stress derivation algorithm is delegated to. The python programming language was selected due to its competency and features for performing calculation on data which was a much needed resource for the stress derivation algorithm. Additionally, python provides an easy means by which to establish server solutions for enabling connectivity from external sources. Python provides easily integrated libraries for establishing TCP/IP based communication links between sources for streaming data from a client connection to the server, like sockets. In this case, the pycom microcontroller is able to establish a socket client connection in order to stream its data to the server which can then be evaluated by the stress derivation algorithm.

The python socket server ultimately overwrote the Thingsboard cloud solution approach as Thingsboard did not provide a means by which to persist data that has been processed by its Rule Engine as Thingsboard provides its own platform for visualization which it expects the developer to use.

5.1.3 React Front-End

A React web application was implemented, as opposed to a React Native mobile application for the dog monitoring system for ease of evaluating a visualization system based on the stress derivation algorithm as the system was evaluated in strictly stationary conditions due to the IoT device monitoring limitation. React is similar to React Native in that it provides a plethora of libraries by the React community providing feature-rich functionality to new applications out of the box. This ease of introducing functionality to the web applications written in React allows for streamlined development of powerful monitoring clients for the purposes of analysing and visualization of processed data in a meaningful manner to the end user.

Notably, React was chosen as the platform for the visualization system of the dog monitoring system for the Google Maps API library provided to React based web applications, that enables the use of the Google Maps Platform to create personalized map-based visualization systems for evaluating the stress derivation data with relation to location.

5.1.4 Flask REST Back-End

A REST-based back-end solution was necessary in the implementation of the dog monitoring system in order to bridge the communication gap between the React-based front-end web application and the socket server as React web applications have no capabilities of communicating over TCP/IP connections. A low-profile Flask back-end was introduced for the python socket server processor to issue the derived stress data via HTTP requests, which the flask back-end could persist for the React web application to access for visualization purposes with the Google Maps API.

5.2 Risk Assessment

5.2.1 Pulse Sensor Usage Conditions - Mild

As the Pulse Sensor is shown to have a notable impact on accuracy of ECG monitoring with relation to pressure changes between itself and the surface it is monitoring, it is important to evaluated and ensure what level of pressure can be applied on the Pulse Sensor before readings become too unstable or noisy for the purpose of evaluating stress from the HR. The Pulse Sensor shows very obvious challenges in stable monitoring during freely mobile conditions but even in the cases of stationary monitoring, it is imperative to note what movements may cause the accuracy of the readings to plummet.

5.2.2 Network Connectivity - Moderate

The IoT dog monitoring system relies on a stable connection due to the constant stream of data output from the Pycom module to a processor server via socket connection. The socket connection relies upon a stable connectivity layer involving an internet connection with sufficient bandwidth. It is important that the IoT Pycom device as well as the processor server are provided a network that supports stable bandwidth and connectivity for a stable stream of data throughout the system.

5.2.3 Application of Pulse Sensor to Dogs - High

Although the Pulse Sensor is designed to capture ECG from a given source, it is not universal in its design. The Pulse Sensor was originally designed for monitoring ECG of humans, and monitoring the HR of a human does not pose nearly as many biological obstructions as it would for dogs. In canine HR monitoring, necessary invasive procedures are typically performed in gold-standard HR monitoring procedures in order to remove obstructive layers of fur. Furthermore, thick layers of flabby/fatty skin can further dampen the signal, for which conductive gel is applied to strengthen the signal and in turn improve overall accuracy. As the propose implementation of the dog monitoring system intends to be non-invasive, necessary precautions must be made to identify an area on the dog's body where there is minimal fur and flabby/fatty skin.

5.3 Implementation Plan

The dog monitoring system implementation plan is involved in two steps. The first step is to have a platform that can provide management and task delegation capabilities in order to establish a structured and catalogued implementation progression plan for the remaining and completed tasks over the course of the implementation of the system.

The second step is to configure a persisted and versioned codebase of the dog monitoring system as a means of documenting the progressive changes in the various aspects of the codebase for the overarching system with relation to new features added over the course of development of the proposed dog monitoring system. This establishes a controlled and maintained means for development of the system.

5.3.1 Gitlab

Gitlab provides a solution to document and version the dog monitoring system codebase which helps in monitoring and managing updates of the system through committing and pushing new changes into the codebase with commentary highlighting what changes have been made. These changes are maintained historically which allows for evaluating the state the system was in before, and what stage of development it is in now.

Gitlab also provides branching of the codebase which can be used to delegate each section of the dog monitoring system an independent area within the codebase for adding updates and documenting the updates for the independent parts of the system. The plan is to use this feature to provide a section for the IoT module, processing server, back-end, and front-end, in order to separate the documentation of updates for each aspect of the system in an orderly manner over the course of development. This documentation subsequently allows review of new code as part of each new commit, permitting extensive review of changes made to the implementation of the system.

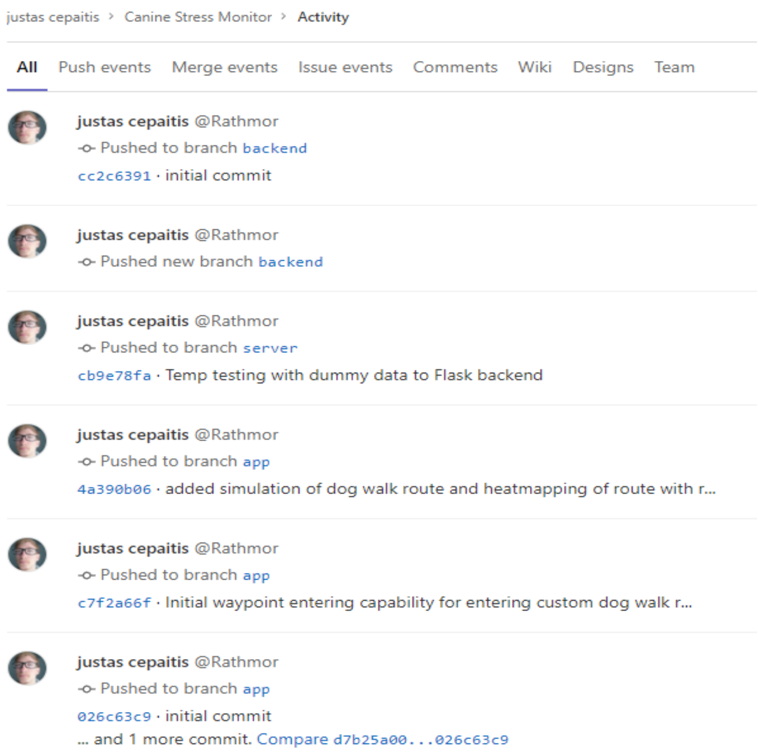


FIGURE 5.4: Image illustrating activity management feature of Gitlab

5.3.2 Trello

Trello is used as part of the implementation of the dog monitoring system to manage and delegate tasks to be completed for the implementation of the system in an iterative and informed manner. The Trello board establishes a clean base of operations for determining the next course of action in developing a system whilst documenting tasks remaining, as well as tasks completed. The board acts as a hub of operations for the course of the implementation and important resources can be catalogued in an agile manner over the course of development, which enables consecutive workloads to remain well defined, and future tasks to be well informed.

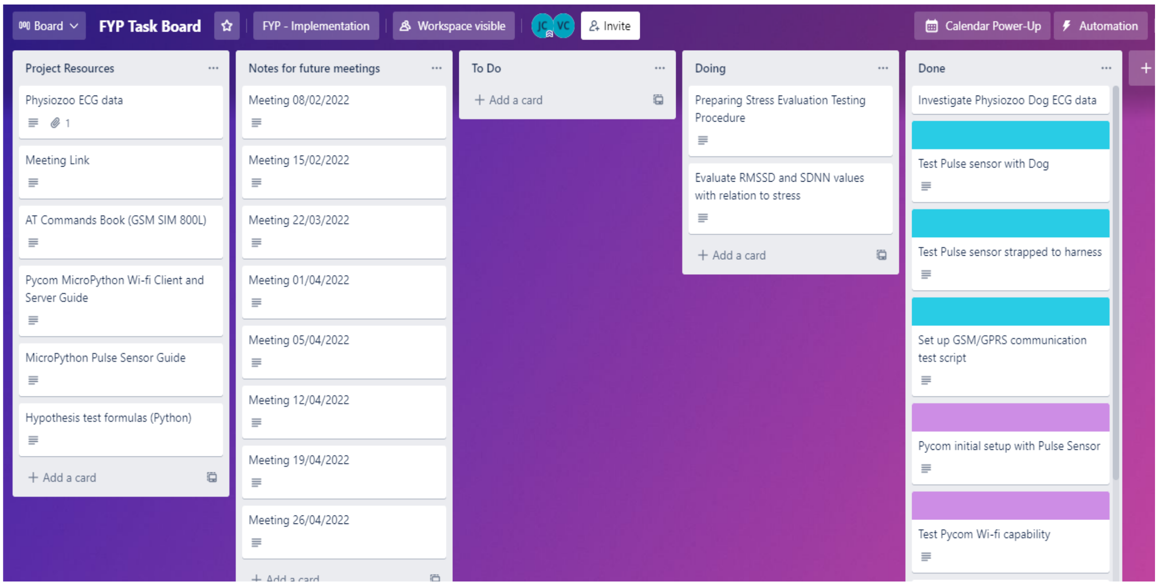


FIGURE 5.5: Image depicting a Trello task management board

5.4 Evaluation

This section enumerates the dog monitoring system stages in development in order to identify and catalogue the main contributions that are desired of the final solution. Furthermore, these stages in development are concise indicators that aid in identifying the level of desired functionality that the system has achieved during the course of development of the dog monitoring system, and what remains to be desired of the system in its stages of development.

- IoT device can read an ECG.
- IoT device can pre-process the ECG for a discernible HR.
- IoT device can extract the necessary HRV metrics (IBI, and successive differences of IBI) from discernible HR for calculating SDNN and RMSSD
- IoT device can establish connectivity and send data to server.
- Server can establish connectivity and access data broadcast from IoT device.
- Server can evaluate and process baseline of HRV data from IoT device.
- Server can derive stress from data broadcast by IoT device.
- Back-end can receive data from server, store data, and send data to front-end.
- Web application can access stored stress derivation data from back-end.

- Web application can simulate an itinerary between an origin and destination provided by the user.
- Web application can create a coloured overlay over the provided itinerary with relation to real-time stress derivation data.

5.5 Methodology

This section enumerates the technical and methodical approach to solving the primary contributions of the proposed dog monitoring system with relation to the supposed approach for deriving stress based on HRV from literature, and providing a map-based visualization system for monitoring the stress of a dog with relation to its location in order to identify potential stressors for the dog.

5.5.1 HRV Based Stress Prediction

HRV is effectively the fluctuation in the time interval between heart beats in an ECG. HRV is comprised of many metrics that can be used to evaluate this fluctuation [26], but the primary metrics of HRV as part of evaluating stress levels are the SDNN and RMSSD. In identifying stress based on HRV, a baseline of RMSSD and SDNN must be established from an ECG distribution, in which the subject being monitored is considered to be in a neutral psychological state. Subsequent processed SDNN and RMSSD metrics from an incoming ECG can then be compared against the baseline in order to identify certain combinations of changes in these values, which can subsequently identify 3 forms of stress: low, moderate, or high.

As the primary contribution of this system is the stress derivation algorithm based on HRV, it is important to highlight the referenced literature that contributed to the development of the stress derivation algorithm, and how the approach may have subsequently changed to meet the needs of the proposed dog monitoring system with relation to evaluating stress in real-time.

5.5.1.1 Heart Rate Variability Predicts the Emotional State In Dogs

This paper [7] was a study performed on a set of dogs in order to evaluate stress in the dogs based on the fluctuation of their HRV metrics, namely the RMSSD and SDNN, in order to derive forms of stress in the canine from the significant change in said metrics. The studies showed that SDNN exhibited significant change during low-stress scenarios, whereas RMSSD exhibited significant change during high-stress scenarios. These significant changes with respect to the other value was that either of these values should be the only ones exhibiting significant change in the cases of low/high stress respectively. Furthermore, the study highlighted that the IBI showed no change during either case of stress derivation. For the stress derivation algorithm, the paper made use of a Wilcoxon signed-rank test in order to evaluate significant change between a baseline of RMSSD and SDNN values collected from a set of dogs (19 dogs for RMSSD, 15 dogs for SDNN) where the dogs were perceived to be in a neutral psychological state at the time of study. This baseline set of SDNN and RMSSD was subsequently compared against two other sets of SDNN and RMSSD values, each derived from scenarios where the canine was in a positive or negative emotional state.

The basis of HRV based stress prediction is formed based on the findings of this paper. One primary concern raised by the proposed solution for deriving stress from HRV in dogs by the approach in this paper is that the solution provided no quantifiable measure as to how much either RMSSD or SDNN should change with relation to evaluating specific stress conditions.

5.5.1.2 Stress Monitoring System Based On Heart Rate Variability Of Dog

This paper [4] was a practical implementation of a system for evaluating a dog's stress levels in real-time through the use of the solution for evaluating stress levels based on HRV provided by the initial paper [7]. The initial approach of the dog monitoring system was to follow the solution approach presented by the stress evaluation system provided by this paper.

Upon further investigation into its implementation, it was found that the paper did not exhibit a correct execution of the Wilcoxon signed-rank test in comparing a baseline of SDNN and RMSSD values against an incoming set of SDNN and RMSSD values for evaluating stress based on HRV. As their proposed solution is evaluating stress in real-time, they must collect a set of SDNN and RMSSD values per 30 or more IBI values extracted from incoming ECG. This means that in their Wilcoxon signed-rank test for evaluating significant change in SDNN and RMSSD against a baseline, they are mixing varied distributions of HRV where the dog may have felt a number of ways with relation to stress, as part of the significant change test against the baseline which likely yielded random results not entirely in line with how the dog may have felt in real-time. As per the literature, the stress derivation algorithm evolved to suit the needs of evaluation based on a real-time basis. As both papers revolved around using the signed-rank test, this form of evaluation was best suited in the late stages of the study, where a large set of ECG readings have already been collected over a time window, and this data, namely the evaluated SDNN and RMSSD, is tested for significant change through a Wilcoxon signed-rank test. This form of evaluation is quite time consuming as a result of the necessity of collecting multiple baselines of SDNN and RMSSD values over 5 minute intervals (or greater), as well as performing the same lengthy evaluation to extract the appropriate SDNN and RMSSD values for the positive/negative testing scenarios. This form of evaluation ultimately only suits to an accurate degree after long-term evaluation of HRV metrics due to the Wilcoxon signed-rank test.

The alternative approach for the stress derivation algorithm for the proposed system shares similarities with the literature, except for evaluating change through the Wilcoxon signed-rank test as it requires a large set of data before it can perform any calculations, which is unsuitable for real-time evaluation. Instead, the IBI and successive differences are collected as part of the baseline, and subsequent necessary metrics are evaluated from this baseline ECG distribution. Namely the SDNN, RMSSD, and mean IBI as the paper [7] found these metrics to be indicative of evaluating stress in dogs.

5.5.1.3 RMSSD and SDNN

Pre-processing must be performed on the incoming ECG in order to extract the necessary values that are used in the evaluation of SDNN and RMSSD through their formulas (See Chapter 2, Figure 2.4). These values are the IBI for SDNN, and the successive differences between consecutive IBI for RMSSD.

The IBI (otherwise known as RRI) stands for the time interval between two discernible beats of the heart in an ECG (See Chapter 2, Figure 2.5). This value is typically denoted in the millisecond (ms) format. As the SDNN metric is effectively the standard deviation of the IBI intervals, it is a time-series metric. As such, it requires a set of IBI values in order to derive an instance of SDNN.

The Successive difference between consecutive IBI values is the time difference between the most recent IBI value, and the previously recorded IBI value. The RMSSD metric is the root-mean square of successive differences. Like SDNN, RMSSD is also a time-series metric, requiring a set of successive differences in IBI in order to derive an instance of RMSSD.

5.5.1.4 ECG Pre-processing

For gathering an ECG, the Pycom FiPy ESP32 controller is connected with the Pulse Sensor module via a serial pin, with which the Pulse Sensor provides the microcontroller a constant serial output stream. The number of this serial output is configured with a threshold of 0 to 1024. These values, if provided with a surface exhibiting a heartbeat, peak when a heartbeat is detected, and the values drop if no heartbeat is found.

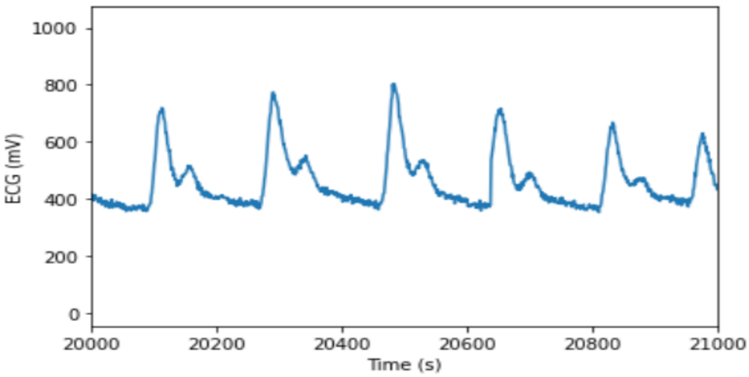


FIGURE 5.6: Visualization of typical serial output from Pulse Sensor when provided a heartbeat.

As the serial output is constantly fluctuating due to the chaotic nature of a heartbeat strength, the pre-processing of the incoming ECG data is required to be dynamic in nature, responding to changes in serial output strength in order to be able to constantly detect a heartbeat, no matter what serial output values are provided at any given time. In order to detect a HR in an ECG, a thresholding algorithm was implemented. The pycom module processes the lowest and highest signal from a window of 250 serial values in order to identify the immediate distribution of ECG, this window of serial values is capped at 250 values, and is constantly being refilled with the new immediate ECG readings in order to keep up with the chaotic nature of the ECG stream. Once these minimum and maximum ECG readings are established, two cut-off points are created, in order to create a "green zone" in the immediate ECG distribution that, when crossed by an incoming ECG, can be identified as a peak in the ECG, which is consistent with a heartbeat.

Any ECG reading from the current distribution of serial values landing above, or on the on-threshold is considered to be a heartbeat. The heartbeat, once identified after passing the on-threshold, is not considered to have concluded until the signal passed below the off-threshold point, which is consistent with the end of a heartbeat.

After establishing a means by which to identify individual heartbeats in an ECG, the IBI and successive difference between IBI values can be derived. These values are necessary

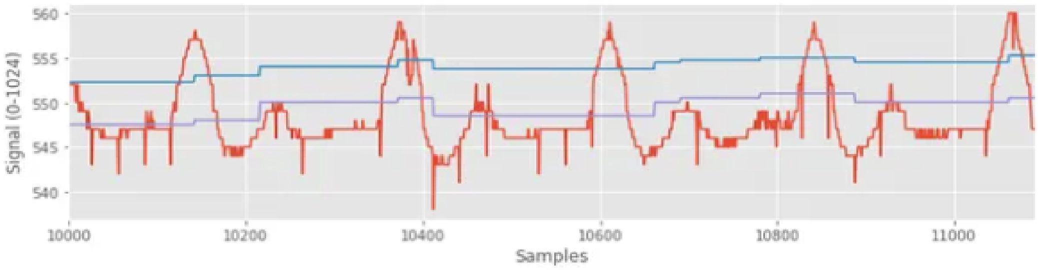


FIGURE 5.7: Illustration of an ECG with cut-off points for on-threshold (blue) and off-threshold (purple).

before the data can be issued to the server for processing the HRV of the ECG into stress as the SDNN and RMSSD require these metrics in order to be evaluated. The IBI is calculated by identifying the time interval between the most recently detected heartbeat in the incoming ECG, and the previously detected heartbeat. When the on-threshold and off-threshold is passed by incoming ECG, it fires a on/off heartbeat value which triggers a timer that begins at the end of the first heartbeat detected, and ends at the start of the next heartbeat detected, before resetting to calculate subsequent IBI. The resulting algorithm provides a time interval between each consecutive set of heartbeats.

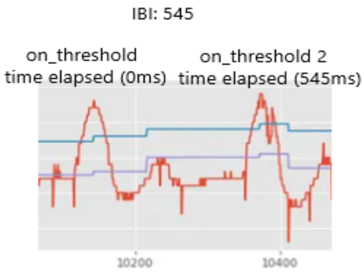


FIGURE 5.8: Illustration depicting the process for evaluating IBI values from an ECG.

For calculating the successive difference between IBI, this process is combined in tandem with the process of calculating consecutive IBI intervals from the incoming samples of ECG. If the previously calculated IBI is logged with regards to the most recently calculated IBI value, the successive difference can be calculated in conjunction with the derivation of IBI values.

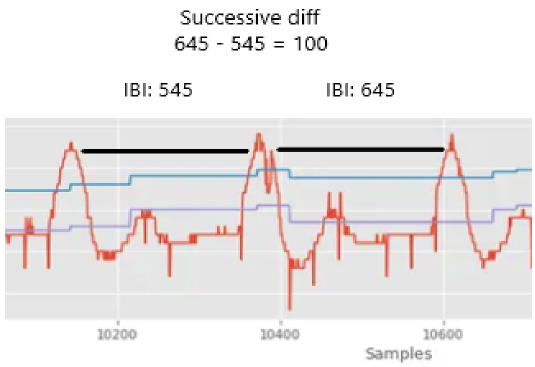


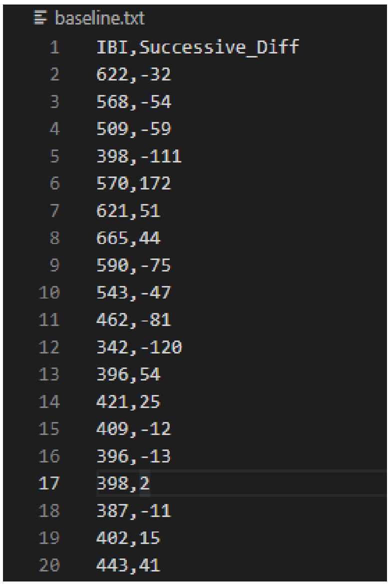
FIGURE 5.9: Illustration depicting the process for evaluating successive differences in IBI from an ECG.

The pre-processed data can then be issued via the Pycom FiPy ESP32 microcontroller's Wi-Fi connectivity, over a socket-client connection to the socket server processor, where the data is processed through a stress derivation algorithm.

5.5.1.5 Baseline Collection

In the significant change comparison for evaluating stress based on HRV, a baseline of metrics needs to be collected as a basis of comparison in which the subject being monitored is in a neutral psychological state. Subsequent real-time ECG monitoring can then be calculated for the relevant SDNN, RMSSD, and IBI metrics in order to perform a significant change comparison against the baseline which derives a low-level stress. As HRV is quite chaotic in nature, it is expected to differ from person-to-person as well as day-to-day due to various factors such as exercise or resting, which often changes HRV as a byproduct. As such, a new baseline should be evaluated in an ideal stress derivation scenario, in order to maximize accuracy.

A python-based socket server processor handles the retrieval of data from the pycom module. This data is streamed over a socket connection between the socket server and the client, which provides the server with a constant stream of IBI and successive difference of IBI values from the IoT device. The python server provides a file-based storage in the event of processing, and persisting a new baseline of HRV related values for when the subject is in a neutral psychological state. The values collected as part of the baseline are the incoming IBI and successive difference between IBI values. The baseline is collected over the course of 5min interval, after which the processor server begins deriving stress. Once a baseline HRV is collected, the relevant HRV metrics of mean IBI, RMSSD, and SDNN are calculated and persisted from the baseline of IBI and successive difference between IBI. These HRV metrics provide a basis of values with which to pair the incoming calculated HRV metrics against in order to identify the varied combinations of significant change, to evaluate a form of stress.



baseline.txt

```
IBI,Successive_Diff
622,-32
568,-54
509,-59
398,-111
570,172
621,51
665,44
590,-75
543,-47
462,-81
342,-120
396,54
421,25
409,-12
396,-13
398,2
387,-11
402,15
443,41
```

FIGURE 5.10: Illustration depicting baseline file structure and the values associated with the baseline.

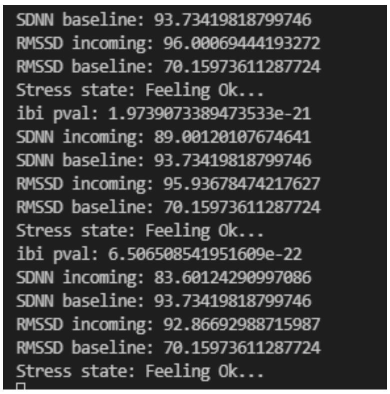
5.5.1.6 Stress Derivation Algorithm

As per literature [7], the mean IBI of the baseline compared against the incoming HRV IBI, from which stress is to be derived, is expected to not have a significant change in either case of significant change in SDNN, which identifies a positive stress, and or RMSSD, which identifies a negative stress. As such, the stress derivation algorithm first identifies the mean IBI of the baseline, and subsequently examines the SDNN and RMSSD metrics of the baseline, against the SDNN and RMSSD processed from the incoming ECG in order to elicit a derivation of stress.

After the processor server operating the stress derivation algorithm has collected a baseline, it has now calculated the mean IBI, SDNN, and RMSSD metrics from the baseline, which can be compared against incoming processed ECG data. Before performing a significant change test, the processor server first collects a set of 30 IBI and Successive Difference values from the broadcast, as SDNN and RMSSD are time-series HRV metrics, and require a window of data to be derived. After 30 values have been collected, the server processes the SDNN (by processing the 30 IBI values), and RMSSD (by processing the 30 successive difference values). this set of 30 values is constantly updated with new incoming IBI and successive difference values from the module, for evaluating incoming HRV, to derive stress on a real-time basis.

For evaluation, a series of hypothesis tests are performed. Starting with the incoming IBI, the mean IBI of the baseline and the set of incoming IBI values from the module are tested in a one-sample T test. This one-sample T test checks the population of incoming IBI values (30 IBI values), and tests if the average IBI of these values is close to the baseline, or not. This hypothesis test ensures that the incoming IBI distribution is no significantly different to the mean IBI of the baseline, which passes the first test of the stress derivation algorithm. Now, the RMSSD and SDNN can be investigated for significant change to perform the final evaluations.

As the SDNN and RMSSD test is involving single values, hypothesis testing cannot be performed as the population of values is not sufficient in volume. As such, a closeness test was performed to determine how close the baseline SDNN or RMSSD is to the incoming processed SDNN and RMSSD from the window of IBI and successive differences. This was performed using the math library in python, using the isclose function. The isclose function is a boolean test that, based on an relative tolerance value for change (0.05 in this case), identifies whether or not the values are close enough relative to each other in value. This function was used to evaluate that the resulting incoming SDNN and RMSSD did NOT appear to be close relative to the baseline SDNN and RMSSD values. If SDNN/RMSSD was found to not be close, relative to RMSSD/SDNN which was close, the stress derivation algorithm would evaluate a positive/negative stress state. In any other case, the stress derivation algorithm would evaluate an average, or OK stress state.



```
SDNN baseline: 93.73419818799746
RMSSD incoming: 96.00069444193272
RMSSD baseline: 70.15973611287724
Stress state: Feeling Ok...
ibi pval: 1.9739073389473533e-21
SDNN incoming: 89.00120107674641
SDNN baseline: 93.73419818799746
RMSSD incoming: 95.93678474217627
RMSSD baseline: 70.15973611287724
Stress state: Feeling Ok...
ibi pval: 6.506508541951609e-22
SDNN incoming: 83.60124290997086
SDNN baseline: 93.73419818799746
RMSSD incoming: 92.86692988715987
RMSSD baseline: 70.15973611287724
Stress state: Feeling Ok...
```

Figure 5.11: Illustration depicting stress derivation algorithm resulting derivation, with depiction of incoming and baseline SDNN and RMSSD evaluations, as well as an associated stress derivation

5.5.2 Visualization System

The visualization client is intended to be an additional layer of analytics for stress derivation in dogs, that can be tied in with an end-user, in order to leverage the valuable insights provided by derivation of stress in a way that seems meaningful to individuals that may wish to know how such a system can be beneficial to them. The visualization client, powered by Google Maps API, is a React-based web application that provides simulation of an itinerary, with heat-mapping capabilities that identifies where over the course of the traversal of the itinerary the subject being monitored may have been stressed. This form of insight can be invaluable to individuals attempting to discern the temperament of a dog as well as in identifying location-based stressors.

The core feature of the React front-end is the Google Maps API's Directions Service API, which provides a generated waypoint between an origin and a destination location that can be found on Google Maps. This API returns an itinerary array of latitude and longitude co-ordinates pertaining to the generated waypoint. the React analytics client processes the generated waypoint by simulating a moving marker across the itinerary's location points, as though an actual device location is being detected as it moves. The movement of the marker is timed in several seconds so as to not move across the point at a speed that is consistent with actual device location movement across a map.

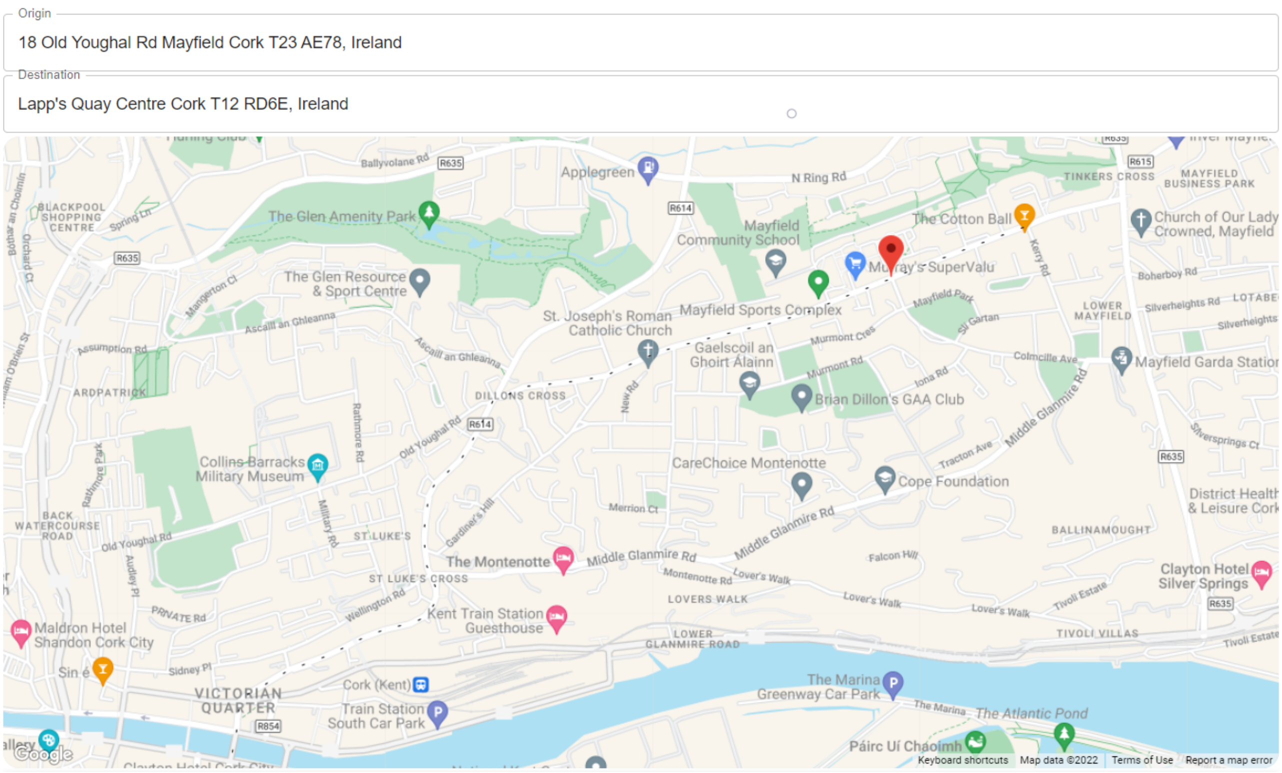


FIGURE 5.12: Illustration showing the Google Maps API generated itinerary (marked by a dotted line) with a simulated marker moving across it

As the marker moves across the itinerary, a Flask back-end bridges the communication on two ends between the socket server processing the stress data, and the react front-end analytics client. The back-end provides two endpoints for storing and retrieving the data, which enables the React front-end visualization client to integrate the stress derivation data into the itinerary simulation, in order to trace a coloured heat-map as the marker moves along the track. The heat-map coloured at a rate of a pre-determined number of latitude and longitude points that the marker traverses. If this value is adjusted, the rate at which the heat-map is drawn can be lengthened, or reduced to a per-point basis which can make the heat-map drawn less/more defined and dense.

The Flask back-end persists the stress derivation data stored by the processor server until the React front-end fires a GET request upon traversing the specified number of points across the itinerary. When the analytics client requests this data, it receives a number of stress derivations in correspondence with the rate at which the heat-map is drawn (ex. the analytics client requests stress derivation data for every 4 points of the itinerary that were traversed). The client then evaluates the stress derivation data to identify which stress state was the most prevalent. Based on this evaluation, the number of points traversed before the request was issued are coloured with the corresponding colour of the most occurring stress state (yellow for moderate stress, green for low stress, and red for high stress) during the traversal of those points.

The Google Maps API provides a polyline feature that creates an overlay over the map, spanning a set of latitude and longitude points. These lines are also be coloured in order to create a heat-mapped line, traced along the latitude and longitude co-ordinates of the itinerary provided by the Directions Service API, that has been passed by the marker.

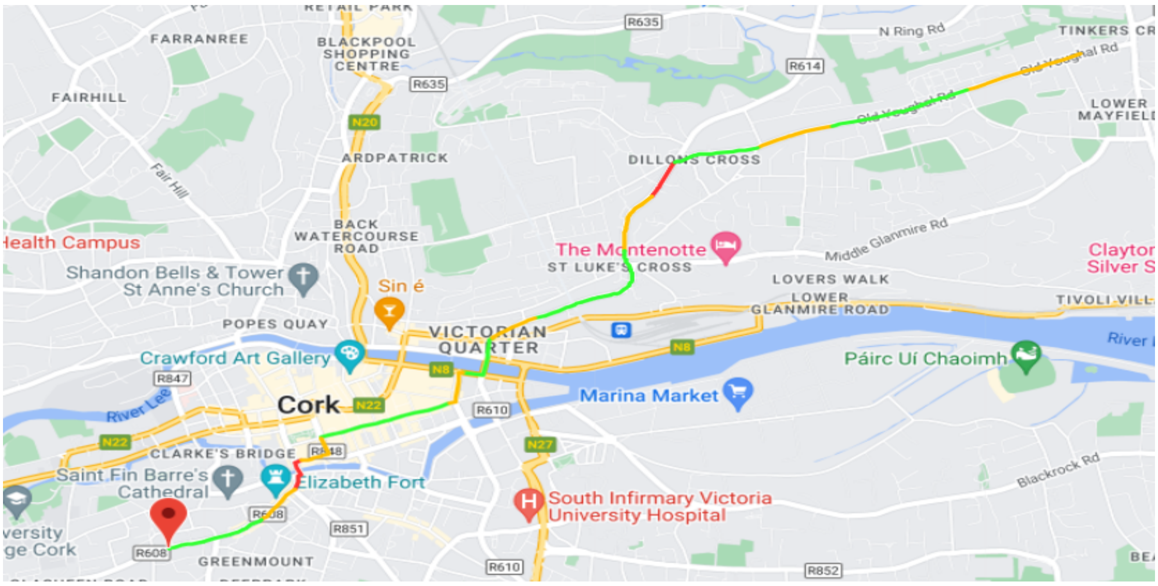


FIGURE 5.13: Illustration showing the stress heat-map polyline, traced over the itinerary provided by Google Maps Directions Service API

5.6 Dog Monitoring System Prototype

The section provides a semi-high level illustration of the to-be prototype of the dog monitoring system based on the above architecture and methodology detail. The illustration is provided in the form of a Data-Flow Diagram (DFD), providing a simplified overview of the machinations of the overall dog monitoring system and the contributions that each aspect of the system provides to the primary contributions of the system which are to have means by which to derive stress based on ECG data from the Pycom module, as well as the front-end analytics client that provides the necessary visualization features using the Google Maps API to evaluate the location of the dog with relation to the derived stress data.

The DFD, although very clearly missing the inner-workings of the algorithms and processes, only seeks to provide a detailed but simplified illustration of the system and what is generally expected of the proposed dog monitoring system implementation with relation to its overall architecture.

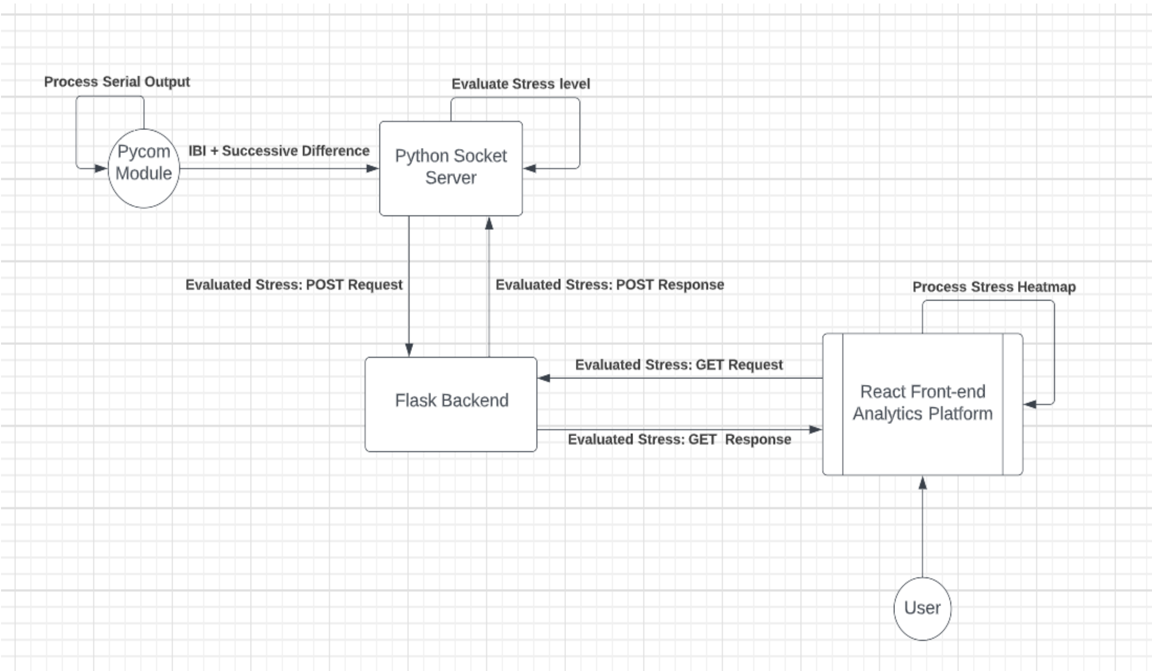


FIGURE 5.14: Data-Flow Diagram (DFD) illustrating the proposed dog monitoring system architecture

5.7 Difficulties Encountered

The evaluation of data in a real-time manner is fraught with challenges as a result of the chaotic nature that analysis of real-time data presents. The primary challenges faced with the implementation of the dog monitoring system was centered around the regions of hardware issues, cloud solutions, HR data cleaning and pre-processing, and the development of a real-time stress derivation algorithm.

- **Easy**: While the Thingsboard cloud solution provided Cassandra database instances for storing data, it only provided this data store for the raw data being output from the devices connected to the Thingsboard Cloud via MQTT. In this proposed dog monitoring system, the task of processing the data in the stress derivation algorithm was to be delegated to the Thingsboard Rule Engine in order to minimize workload on the IoT module, the cloud solution did not however provide a means to store processed data.

  The Thingsboard Cloud was scrapped in favour of a local socket server processor that performs the necessary processing of stress from received metrics via socket connection from the module.

- **Medium**: Initially, the approach for the IoT module was to use an Arduino in combination with the Pulse Sensor due to the Pulse Sensor providing a library for Arduino solutions that performs all the necessary operations to process the serial output from the Pulse Sensor, and calculating the necessary metrics for the stress derivation algorithm. However, the Arduino Uno module used did not have a means by which to issue the processed data remotely and the GSM module for establishing remote communication had a too weak of a signal strength.

  A Pycom module was used in place of the Arduino Uno, and the necessary pre-processing of the serial output from the Pulse Sensor had to be developed as the Pulse Sensor provided no libraries written in MicroPython. A relevant resource was found on how to pre-process Pulse Sensor serial output in MicroPython [41].

- **Medium**: After performing multiple evaluations on the Pulse Sensor and its ability to achieve a regular reading from the dog's pulse without jitters, the readings were found to have too many blank spots of incoming data due to flat-line data with no identifiable HR. The evaluation involved the dog in various positions (at its leisure), such as sitting, standing, lying down. The evaluation yielded that the sensor was only able to get a stable ECG reading when the dog stood a certain way and also if their heart-beat was strong enough. The limitations invoked by the lack of invasive procedures combined with a commercial HR sensor not inherently designed for dogs made it unreliable to perform further evaluation of the dog monitoring system using live readings from a dog.

After additional investigation into literature [42] [7] [4], research showed that the HRV metrics used in the stress evaluation procedure have near identical responses to stress in humans as in dogs. As such, further evaluation of the stress derivation algorithm was performed using a human's HR.

- **Medium**: During the implementation of the stress derivation algorithm, a referenced paper [4] provided a detailed approach for processing the HRV RMSSD and SDNN metrics in real-time in order to derive from them, 3 types of stress. This paper followed the guidelines instructed by another paper [7] which provided instructions as to how this stress derivation can be achieved through collecting a baseline of processed RMSSD and SDNN values when the subject dog was in a neutral psychological state, which was then used as a basis of comparison in a signed rank test against RMSSD and SDNN values processed from incoming real-time ECG data. However, the baseline of RMSSD and SDNN values was a cumulative baseline of neutral psychological states collected from multiple dogs, and not just one as was the case of this approach.

  The approach for the stress derivation algorithm was reassessed as the signed rank test cannot be performed with just a single set of RMSSD and SDNN baseline values (see Chapter 2, Figure 2.7). Instead, collecting the incoming IBI and Successive Difference values as a baseline in place of the RMSSD and SDNN in the literature allows us to evaluate the incoming set of IBI values against the baseline IBI values to determine significant change in the ECG distribution. This form of evaluation for significant change in place of the signed rank test was eligible, as RMSSD and SDNN metrics are derivatives of IBI. After this procedure, the incoming single set of SDNN and RMSSD from baseline can be compared against the RMSSD and SDNN processed from the baseline of IBI and Successive Difference in IBI to evaluate significant change.

- **Hard**: The Pulse Sensor used posed a hardware limitation in reading stability with relation to pressure changes between it and the surface being monitored for an ECG. Slight pressure changes between the Pulse Sensor and the surface would dampen the serial output from the Pulse Sensor to the point of a flat-line ECG which resulted in an indistinguishable pulse. These changes in pressure made the device unsuitable in ambulatory conditions due to the movement of the surface the sensor is placed against causing occasional pressure changes between the surface and the sensor due to movement, muscle contractions or otherwise.

  The goal of collecting location data from a real device to monitor the movements of a dog was re-assessed, and a simulation of movement across an itinerary provided by the google maps API was performed in its stead as a means of proof of concept.

Chapter 6

Testing and Evaluation

6.1 Evaluation

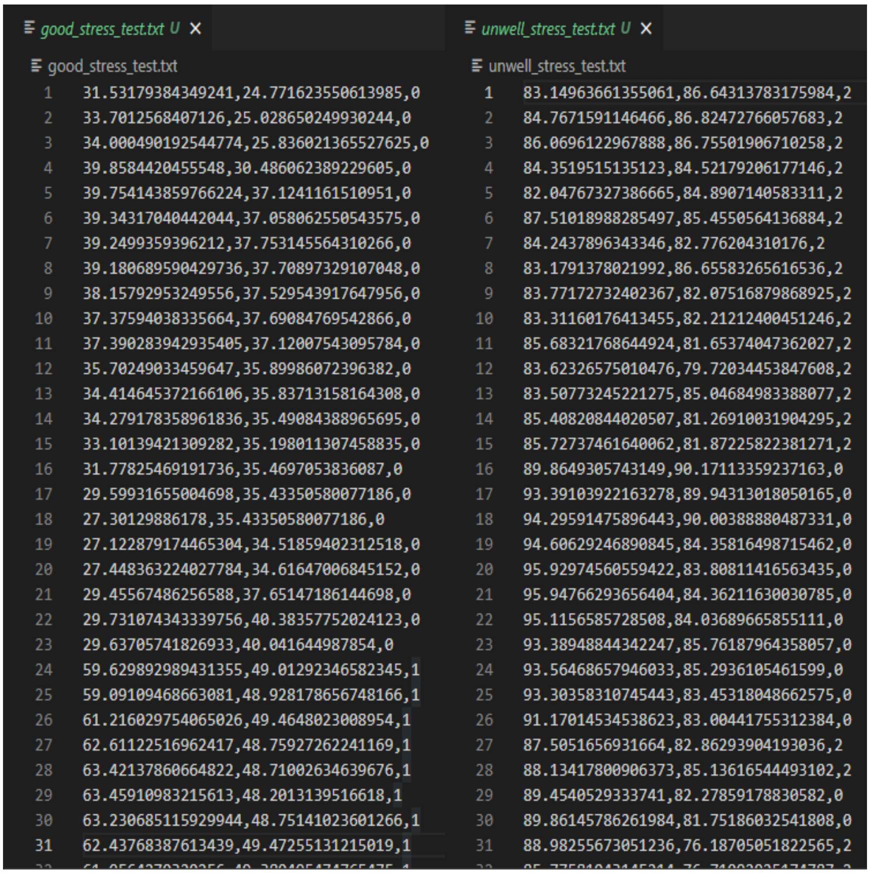
The primary contribution of the proposed dog monitoring system was to identify how stress in a subject can be derived based on HRV. As the subsequent features of the monitoring system are dependent on the results from the stress derivation algorithm that analyzes HRV metrics to derive stress (ex. the visualization client), the evaluation was focused on identifying the overall level of accuracy of the stress derivation algorithm, performed in real-time with relation to test cases stimulating a specific type of emotional response.

The processor server involved in processing the stress derivation algorithm can be informed through terminal input while running, to begin writing the HRV metric results and the stress derivation into a file during a user-controlled test. As part of this test, while the processor server is outputting the HRV metric and stress derivation values to the file, the subject watches stimulating video material, that elicits an emotional response of happiness or sadness. In these evaluation cases, a file is created for each form of video material, compiling a set of HRV metrics and stress derivations denoting how the subject was feeling when viewing the material. The tests would involve 5 minutes of evaluation where the subject would watch media eliciting a positive/negative emotional response. The concept behind this form of evaluation is to identify, in a real use-case, the accuracy of prediction of the stress derivation algorithm in identifying the expected stress state of the subject when watching the relevant material that should, in an ideal scenario, have made them exhibit a certain form of emotional response. It is expected that the subject would feel less stressed, and better, when viewing positive material. Alternatively, the subject is expected to feel more stressed, and less well, when viewing negative material.

The basis of the evaluation is to quantify the HRV metrics as part of the tests that intend to stimulate a certain emotional response, and correlate those values, along with the derived stress, the overall accuracy of the stress derivation, as well as identify the trend of change in the HRV metrics pertaining to the tests.

6.2 Metrics

The metrics involved in the evaluation of the stress derivation algorithm are the incoming SDNN and RMSSD values processed from the time window of IBI and successive differences, as well as the corresponding stress derivation that was evaluated at the time on a real-time basis. These metrics as part of the test conducted for stimulating positive/negative stress response through video material, were stored in files. The SDNN and RMSSD are logged as part of the evaluation in order to determine the quantifiable aspect of the stress derivation algorithm when deriving stress during the specific testing instilling a certain emotional response (good/unwell), and the stress derivations are stored in order to conduct an accuracy examination on the resulting data from the conducted tests.



good_stress_test.txt

```
31.53179384349241,24.771623550613985,0
33.7012568407126,25.028650249930244,0
34.000490192544774,25.836021365527625,0
39.8584420455548,30.486062389229605,0
39.754143859766224,37.1241161510951,0
39.34317040442044,37.058062550543575,0
39.2499359396212,37.75314556431026,0
39.180689590429736,37.70897329107048,0
38.15792953249556,37.529543917647956,0
37.37594038335664,37.69084769542866,0
37.390283942935405,37.12007543095784,0
35.70249033459647,35.89986072396382,0
34.414645372166106,35.83713158164308,0
34.279178358961836,35.49084388965695,0
33.10139421309282,35.198011307458835,0
31.77825469191736,35.4697053836087,0
29.59931655004698,35.43350580077186,0
27.30129886178,35.43350580077186,0
27.122879174465304,34.51859402312518,0
27.448363224027784,34.61647006845152,0
29.45567486256588,37.65147186144698,0
29.731074343339756,40.38357752024123,0
29.63705741826933,40.041644987854,0
59.629892989431355,49.01292346582345,1
59.09109468663081,48.928178656748166,1
61.216029754065026,49.4648023008954,1
62.61122516962417,48.75927262241169,1
63.42137860664822,48.71002634639676,1
63.45910983215613,48.2013139516618,1
63.230685115929944,48.75141023601266,1
62.43768387613439,49.47255131215019,1
```

unwell_stress_test.txt

```
83.14963661355061,86.64313783175984,2
84.7671591146466,86.82472766057683,2
86.0696122967888,86.75501906710258,2
84.3519515135123,84.52179206177146,2
82.04767327386665,84.8907140583311,2
87.51018988285497,85.4550564136884,2
84.2437896343346,82.776204310176,2
83.1791378021992,86.65583265616536,2
83.77172732402367,82.07516879868925,2
83.31160176413455,82.21212400451246,2
85.68321768644924,81.65374047362027,2
83.62326575010476,79.72034453847608,2
83.50773245221275,85.04684983388077,2
85.40820844020507,81.26910031904295,2
85.72737461640062,81.87225822381271,2
89.8649305743149,90.17113359237163,0
93.39103922163278,89.94313018050165,0
94.29591475896443,90.00388880487331,0
94.60629246890845,84.35816498715462,0
95.92974560559422,83.80811416563435,0
95.94766293656404,84.36211630030785,0
95.1156585728508,84.03689665855111,0
93.38948844342247,85.76187964358057,0
93.56468657946033,85.2936105461599,0
93.30358310745443,83.45318048662575,0
91.17014534538623,83.00441755312384,0
87.5051656931664,82.86293904193036,2
88.13417800906373,85.13616544493102,2
89.4540529333741,82.27859178830582,0
89.86145786261984,81.75186032541808,0
88.98255673051236,76.18705051822565,2
```

FIGURE 6.1: Illustration depicting file contents of each stress evaluation conducted. Structure of the data is: SDNN, RMSSD, stress (0 for moderate, 1 for good, 2 for unwell)

6.3 Results

The two resulting files (see Figure 6.1) and their contents were evaluated through pandas, in order to analyse and identify the quantitative aspect of the resulting HRV metrics during either evaluation conducted, that stimulated a certain emotional response. Two plots were manifested from the dataframes generated by pandas, that mapped out the RMSSD, and SDNN values over the course of monitoring during the specific test type, in order to identify the distribution of data. The two plots were generated in matplotlib and identify no drastic difference in distribution during either one of the tests, suggesting that SDNN and RMSSD have a quantitative change that is not differing and the chaotic nature of the values persists in either evaluation case eliciting a certain emotional response.

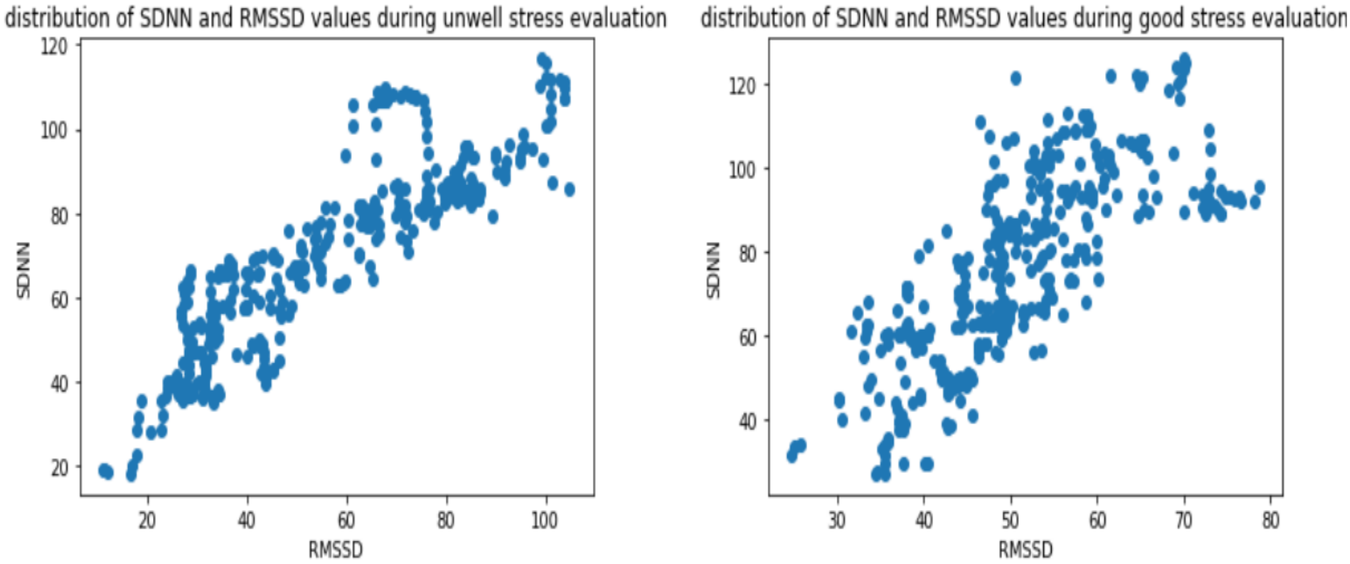


FIGURE 6.2: Illustration depicting graphs that plot SDNN and RMSSD metrics based on the following evaluation types

Additional statistical analysis was performed on the resulting files in order to determine the number of times the overall evaluation reported specific forms of stress, in order to identify a general accuracy of the stress derivation algorithm in either case for the form of evaluation conducted that was evaluating the emotional response of a subject while they viewed positively/negatively stimulating material. The results found that the evaluations much more often than not, reported a moderate level stress (otherwise known as ok stress), but it is also important to note that the resulting evaluations were performed on a real-time basis of the subject viewing the material, and they cannot have been expected to immediately have experienced a specific emotional response, nor can they have been expected to constantly feel such an emotional response over the course of viewing the material. Despite these fall-backs, the results depicted the stress derivation algorithm to correctly derive the stress state in the specific evaluations with regards to deriving the expected stress state, when the subject viewed the varied material based on the evaluation type. The statistical analysis performed was a count of the number of times a specific stress type was reported, and the overall accuracy of the algorithm over the course of the evaluation, to correctly evaluate a stress condition.

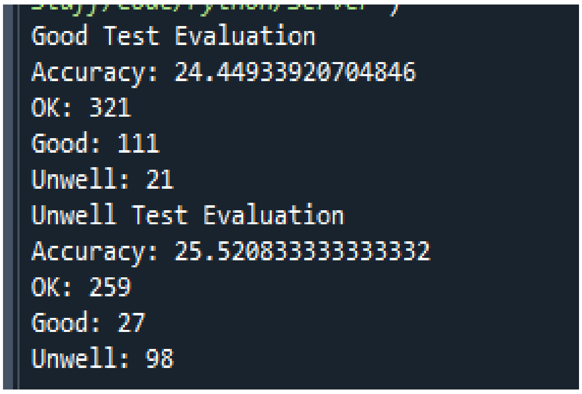


Figure 6.3: Illustration

The resulting evaluation showed clearly that the stress derivation algorithm, was able to relatively appropriately identify the stress state of the subject being monitored, while they viewed the according media that would specifically elicit a positive/negative emotional response. It is important to note in this case however, that the resulting SDNN and RMSSD calculations from the incoming values are based on a window of IBI and successive differences from a pool of 30 values. This pool, if increased, could provide even more accuracy with stress derivation, although this also means that the evaluation of stress will be even more delayed, as the window of values is a past window, which effectively means that the stress derivation derives stress at a slight delay to real-time ANS activity with relation to changes in HRV. The subsequent tests were also performed on a single subject who anticipated the evaluations, which may have had an impact on the resulting stress derivations. Furthermore, as a note on the distribution of SDNN and RMSSD values depicting an increase/decrease in the HRV metrics in the plots, this may be a result caused by the HRV metrics initially being low during the beginning stages of the media, while the subject was establishing focus with the media, in order to elicit a change in their HRV. As the resulting examinations were performed on samples of media over a 5 minute evaluation for either test case, the subject may have been in a "wavy" state, where they would have felt happiness, and return to a neutral psychological state, as subsequent snippets of media were presented to them.

While these examinations were performed on a human subject, literature has suggested that the same HRV metrics used in evaluating canine stress, can be repurposed for stress evaluation in humans. The above evaluation supports these claims as the accuracy has shown relative sense in either test evaluating a subject viewing media that would elicit a positive/negative stress. Based on these findings, the stress derivation system, if paired with more sophisticated hardware that can bypass the obstructive biological factors of canines (fur and thick/fatty skin layers), it can be repurposed for evaluating stress in dogs.

Chapter 7

Discussion and Conclusions

7.1 Project Review

The proposed dog monitoring system, that can derive stress from HRV, was largely involved in fact-checking, research, and primarily the operation of real-time data as the proposed system was largely involved in providing a means by which to evaluate a canine's stress levels in real-time. Over the course of the project, the research was extremely involved within the scientific subject with regards to verifying the appropriate HRV metrics that can be used in evaluating stress, as well as the types of metrics that are appropriate for evaluation in cases of real-time stress derivation. It became clear to me the importance of thorough research, especially within the scope of a system revolving around scientific evaluation.

One of the primary lessons learned over the course of implementing the dog monitoring system was the iterative nature of development. It was important in a research-based project, to ensure that subsequent work was followed through in a manner that was subject to change, which was very evident over the course of my implementation of the dog monitoring system, when the resulting hardware selected for the purposes of non-invasive ECG monitoring of a canine, failed to meet the necessary requirements of relatively accurate HR readings, in which these readings were the basis of the stress derivation based on HRV, and were imperative to the success of the dog monitoring system. The monitoring approach was then subsequently changed to evaluating human ECG, which was backed up by additional scrutiny of appropriate literature.

Evaluating real-time, chaotic data is no easy feat, as depicted by the dog monitoring system, when evaluating the stress derivation algorithm and generally the overall system. Necessary precautions required to be taken to ensure that the stream of readings from the ECG were pre-processed accordingly, and further more accounted for necessary noise cleaning operations in order to ensure that the data presented to the subsequent stress derivation algorithm, and other processes, was presentable and not going to impact overall accuracy. The dog monitoring system implementation showed the importance of cleaning, and maintaining real-time data traffic due to the chaotic nature of the data that may be presented by a real-time monitoring basis.

Despite the shortcomings of the resulting implementation of the dog monitoring system such as the hardware challenges affecting the resulting robustness of the overall system in that visualization was simulated, and the evaluation was performed in stationary conditions, the final stress derivation algorithm cannot be duly noted. Through scrutinized research, the system revealed a means by which to derive low-level stress based on HRV fluctuations at a real-time basis.

From the visualization system, valuable insights were made into how real-time data can be leveraged through a variety of libraries and APIs, to provide an analytical and visually meaningful means by which to evaluate the data, and visualize the way the data may interact with external data, like the real-time location heat-mapping, with relation to real-time stress derivations. The implementation of this analytics client provided valuable knowledge into the areas of data analytics, with how the stress derivation data can be leveraged with visualization systems, in order to extract additional insights from a given set of data, in order to exploit the features provided by differing sets of data, when combined with additional resources.

7.2 Future Work

The visualization client was quite low-profile, in that much more can be leveraged in future adaptations of the dog monitoring system, in order to leverage additional insights from the visualization potential the client already provides through logging, and heat-mapping itinerary co-ordinates with relation to stress derivations. The client could become much more robust with a back-end hosting a database, that can catalogue, and store latitude and longitude co-ordinates of trips, with relation to the stress evaluations heat-mapped at each co-ordinate, which can provide trainers and dog owners a layered visualization system that can provide invaluable insights into records of activities performed by the dog, and the associated stress heat-maps, which can identify trends in the stress of the dog with relation to locations, which could be indications as to what the dog's temperament is, based on the locales that act as stressors to them.

With a more sophisticated hardware implementation, the system could potentially provide stable ECG readings from the dog, even in ambulatory conditions, with which the system can be adapted to use real-time location data to map out itineraries using the current visualization client. This form of usage can potentially expose the system to additional use-cases, that can provide additional insights into a canine's temperament and general demeanour. Owners could use such a monitoring system during play-time activities of the dog in order to identify the dog's stress state in differing conditions, other than moderated walking activities.

The initial inspiration of the dog monitoring system was to aid guide dog trainers in evaluating canine performance during training with foster families. If the hardware limitations imposed by a IoT sensor for monitoring ECG were lifted by more sophisticated hardware, the dog monitoring system could expand upon the visualization system to also include an activity monitor, where foster families can input a certain activity that is currently being performed by the dog, in order to perform evaluation on the dog's performance during a training activity, which can provide status reports of the dog's performance with relation to their stress states over the course of the exercises, and additional health metrics based on ECG, such as their average HR.

Bibliography

[1] N. Sakovich, “Top eight iot platforms to watch in 2019,” June 2019.

[2] H. Tang, J. Shi, and K. Lei, “A smart low-consumption iot framework for location tracking and its real application,” in *6th international conference on electronics information and emergency communication (iceiec)*. IEEE, 2016, pp. 306–309.

[3] R. Brugarolas, T. Latif, J. Dieffenderfer, K. Walker, S. Yuschak, B. L. Sherman, D. L. Roberts, and A. Bozkurt, “Wearable heart rate sensor systems for wireless canine health monitoring,” vol. 16, no. 10. IEEE, 2015, pp. 3454–3464.

[4] H. Y. M. S. Jeonghee Chi, Chaeeun Lee, “Stress monitoring system based on heart rate variability of dog,” vol. 9, no. 2. Journal of Scientific Technology Research, 2020.

[5] A. Taivalsaari and T. Mikkonen, “On the development of iot systems,” in *3rd International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2018, pp. 13–19.

[6] G. Ramesh, K. Sivaraman, V. Subramani, P. Y. Vignesh, and S. V. V. Bhogachari, “Farm animal location tracking system using arduino and gps module,” in *International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 2021, pp. 1–4.

[7] M. Katayama, T. Kubo, K. Mogi, K. Ikeda, M. Nagasawa, and T. Kikusui, “Heart rate variability predicts the emotional state in dogs,” vol. 128. Elsevier, 2016, pp. 108–112.

[8] D. Wormald, A. J. Lawrence, G. Carter, and A. D. Fisher, “Reduced heart rate variability in pet dogs affected by anxiety-related behaviour problems,” vol. 168. Elsevier, 2017, pp. 122–127.

[9] M. Foster, R. Brugarolas, K. Walker, S. Mealin, Z. Cleghern, S. Yuschak, J. C. Clark, D. Adin, J. Russenberger, M. Gruen *et al.*, “Preliminary evaluation of a

wearable sensor system for heart rate assessment in guide dog puppies," vol. 20, no. 16. IEEE, 2020, pp. 9449–9459.

[10] M. Zupan, J. Buskas, J. Altimiras, and L. J. Keeling, "Assessing positive emotional states in dogs using heart rate and heart rate variability," vol. 155. Elsevier, 2016, pp. 102–111.

[11] R. A. Galosy, L. K. Clarke, and J. H. Mitchell, "Cardiac changes during behavioral stress in dogs," vol. 236, no. 5. American Physiological Society Bethesda, MD, 1979, pp. H750–H758.

[12] S. D. Kreibig, "Autonomic nervous system activity in emotion: A review," *Biological psychology*, vol. 84, no. 3, pp. 394–421, 2010.

[13] G. Team, "What is geolocation?" April 2020.

[14] Tractive, "Tractive."

[15] Whistle, "Whistlego."

[16] D. W. Bailey, M. G. Trotter, C. W. Knight, and M. G. Thomas, "Use of gps tracking collars and accelerometers for rangeland livestock production research," vol. 2, no. 1. Oxford University Press US, 2018, pp. 81–88.

[17] G. Chindhe, A. Javali, P. Patil, and P. Budhawant, "A survey on various location tracking systems," *Int. Res. J. Eng. Technol*, vol. 5, no. 12, pp. 671–675, 2018.

[18] M. Specht, C. Specht, A. Wilk, W. Koc, L. Smolarek, K. Czaplewski, K. Karwowski, P. S. Dabrowski, J. Skibicki, P. Chrostowski *et al.*, "Testing the positioning accuracy of gnss solutions during the tramway track mobile satellite measurements in diverse urban signal reception conditions," *Energies*, vol. 13, no. 14, p. 3646, 2020.

[19] M. Tscharke and T. M. Banhazi, "A brief review of the application of machine vision in livestock behaviour analysis," *Agrárinformatika/Journal of Agricultural Informatics*, vol. 7, no. 1, pp. 23–42, 2016.

[20] AADI, "Autism assistance dogs ireland," https://www.autismassistancedogsireland.ie/.

[21] V. S. Jonckheer-Sheehy, C. M. Vinke, and A. Ortolani, "Validation of a polar® human heart rate monitor for measuring heart rate and heart rate variability in adult dogs under stationary conditions," vol. 7, no. 4. Elsevier, 2012, pp. 205–212.

[22] A. Essner, R. Sjöström, E. Ahlgren, and B. Lindmark, "Validity and reliability of polar® rs800cx heart rate monitor, measuring heart rate in dogs during standing position and at trot on a treadmill," vol. 114. Elsevier, 2013, pp. 1–5.

[23] Polar, "Polar."

[24] A. H. Zohar, C. R. Cloninger, R. McCraty *et al.*, "Personality and heart rate variability: exploring pathways from personality to cardiac coherence and health," *Open Journal of Social Sciences*, vol. 1, no. 06, p. 32, 2013.

[25] S. Bogucki and A. Noszczyk-Nowak, "Short-term heart rate variability (hrv) in healthy dogs," *Polish journal of veterinary sciences*, vol. 18, no. 2, 2015.

[26] F. Shaffer and J. P. Ginsberg, "An overview of heart rate variability metrics and norms," *Frontiers in public health*, p. 258, 2017.

[27] Arduino, "Wemos d1 mini."

[28] Thingsboard, "Thingsboard."

[29] L. Frondelius, K. Järvenranta, T. Koponen, and J. Mononen, "The effects of body posture and temperament on heart rate variability in dairy cows," *Physiology & Behavior*, vol. 139, pp. 437–441, 2015.

[30] Wikipedia, "Wilcoxon signed-rank test."

[31] U. Bareth and A. Kupper, "Energy-efficient position tracking in proactive location-based services for smartphone environments," in *35th Annual Computer Software and Applications Conference*. IEEE, 2011, pp. 516–521.

[32] Google, "Google maps."

[33] Amap, "Amap."

[34] P. Kanani and M. Padole, "Real-time location tracker for critical health patient using arduino, gps neo6m and gsm sim800l in health care," in *4th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2020, pp. 242–249.

[35] Arduino, "Arduino uno."

[36] ——, "Tinygps."

[37] Google, "Google maps api."

[38] M. Lampe, J. Bräuer, J. Kaminski, and Z. Virányi, "The effects of domestication and ontogeny on cognition in dogs and wolves," *Scientific Reports*, vol. 7, no. 1, pp. 1–8, 2017.

[39] W. F. Electronics, "Pulse sensor."

[40] Facebook, "React native."

[41] M. Fitzpatrick, "Building a micropython heart rate monitor."

[42] F. Shaffer and J. P. Ginsberg, "An overview of heart rate variability metrics and norms," *Frontiers in public health*, p. 258, 2017.

Appendix A

Code Snippets

Put appendix material in this section e.g. code snippets

USE THE APPENDICES

Appendix B

Wireframe Models