

基於生理感測及機器學習技術之智慧輸液系統

Smart Infusion System based on Physiological Sensing and Machine Learning Technology

¹ 聶羽柔, ¹ 杜孟軒, ¹ 黃郁雯, ¹ 葉旻純, ^{1*} 梁家銘, ² 陳建志, ³ 郭瑾

¹ 國立臺南大學 電機工程學系

² 國立陽明交通大學 智慧計算與科技研究所

³ 國立成功大學醫學院附設醫院 腫瘤醫學部主治醫師

*通訊作者: jmliang.cs@gmail.com

摘要

為因應逐漸升高的護病比與提升病患的照護品質，本研究將針對靜脈注射這項常見的醫療處置，配合物聯網與機器學習，設計一套具有以下三項功能的系統：1.能夠實時呈現患者生理數據的網頁，2.量測與控制輸液速度的裝置，3.以機器學習發出輸液速度建議及發出患者身體異常的提醒。在前端硬體裝置的部分，本研究採用 Arduino 作為前端感測器的物聯網通訊平台，透過無線網路傳送前端裝置測得的數據與接收後端發送的指令。在後端的部分採用 Node.js 架設物聯網與網頁伺服器，達成即時的數據顯示與遠端控制。為了進一步加速患者生理情況之判讀，此系統亦加入機器學習將讀取到的資料進行分析，以幫助護理人員提供患者更快速且精確的醫療處置，透過科技輔助達成精準地輸液照護。

關鍵詞：生理感測、機器學習、無線網路、輸液控制、物聯網

* Corresponding author: jmliang.cs@gmail.com

DOI : 10.53106/222344892022101202004

1. 緒論

靜脈注射（輸液/點滴）是將人體缺乏的水分、葡萄糖、電解質或藥液藉由靜脈留置針補充至體內，對於無法自行進食或病情危急之病患是重要的醫療處置。輸液量與輸液速度通常是依靠醫護人員的經驗判斷，且須時時依據病患之病況做調整，然而，現今平均每一名護理人員要照顧約十名病患，由於醫護人手短缺，常未能及時發現病患須調整輸液。此外，市面上廣泛使用的輸液調整設備為手動調節，無法精確知道輸液速度，因此，精確地給予、並時時對應病況調節輸液，對於護理照護有其困難，綜合以上所述，一套能兼具遠距、同步、且可實時監測病患生理狀態之輸液調節系統，於輸液照護有其必要性。

本研究提出一套智慧輸液系統，其主要是基於 Node.js 平台的物聯網解決方案，在前端裝置的部分採用 Arduino 作為物聯網通訊平台，以市售的生理感測器搭配自行設計之輸液感測器，達成輸液及患者狀態的監看。後端使用 Node.js 與 My SQL 架設物聯網及網頁伺服器，將前端硬體感測之數據同步傳送至網頁。為了加速患者生理情況之判讀，本系統使用機器學習將讀取到的資料進行分析，再透過本系統將判讀結果顯示於網頁中，方便醫療人員在如手機和護理站電腦的介面上即時查看，最終再由我們自行研發的輸液速度控制器來達成遠端調控，甚至是開啟智能模式讓它自行調控。本研究的目標是使此項智慧輸液系統能夠舒緩醫療人力的不足，透過物聯網及機器學習等科技，輔助日後逐漸嚴苛的醫療環境。

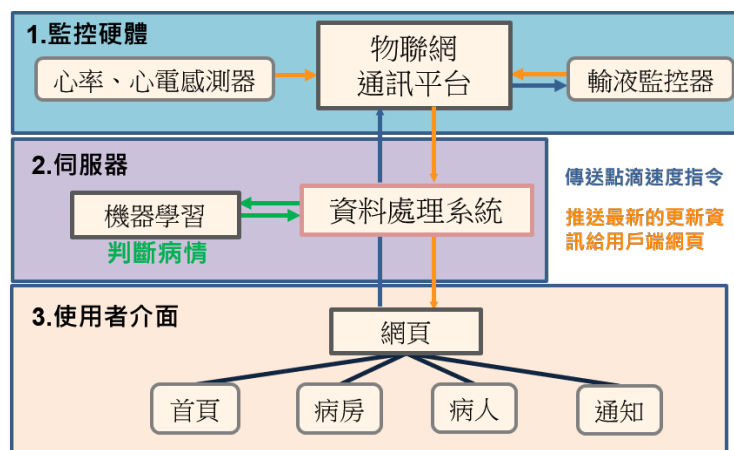
2. 文獻回顧與探討

靜脈注射（輸液/點滴）雖然是一項常見的醫療行為，但是其實存在著不小的風險，尤其是針對小孩、老人及重症病患。輸液量多寡造成的問題[1]，如：輸液量太少：增加脫水的風險，嚴重時會導致腎功能衰竭。反之，輸液量過多：造成低血鈉症等。由上述可知輸液時給予過多或過少的液體都會造成病患的危險，而且當輸液結束卻無即時更換，容易使血液逆流，導致須重新靜脈注射，重複此動作會增加併發症發生的機率。並且因為病人欠缺對藥品的認知，或忽於追蹤輸注部位狀況等，造成護理師要一直關注病患身體狀況。

針對機器學習方面，程式需要展現出：利用現有的經驗（E），不斷改善其完成既定任務（T）的性能（P）的特性[2]，如：讀取儲存在資料庫中現有的資料，經過演算法再從這些資料中自動分析規律[3]以及與不同變因之間的相關性[4]。在圖型識別領域中，KNN 演算法（k-nearest neighbor classification）是一種用於分類和迴歸的無母數統計方法[5]。在未來，當得到新資料時，可以在短時間透過訓練出來的模型去進行預測。

3. 研究方法及步驟

(一) 系統架構與流程



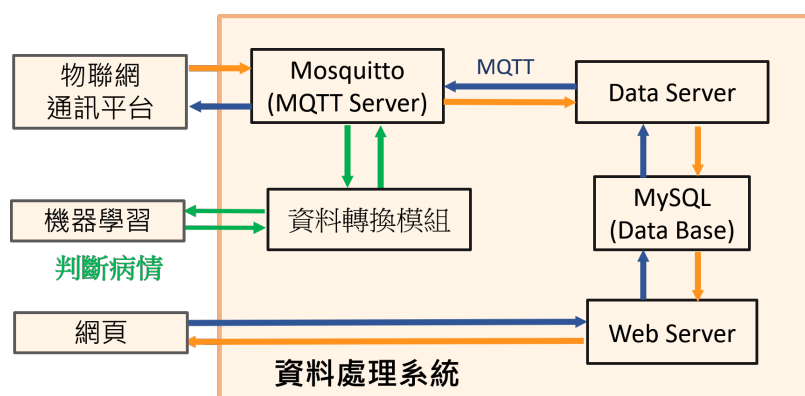
圖一：系統架構圖

1. 監控硬體單元

本研究採用較為簡易的 Arduino Yún 作為物聯網通訊平台，以 MQTT 協定收發感測及控制元件與伺服器之間的溝通。感測元件之心率感測器用於監控相對輕症病人的身體數據，Arduino 會將患者的心率持續回傳至伺服器中，讓護理師能夠簡單了解病人目前的狀況。感測元件之心電感測器用於監測相對重症病患的身體數據，Arduino 將患者的心電數據回傳伺服器並以機器學習判斷有無心肌缺血，並同時於網頁中繪製出心電圖，提供護理人員更詳細的數據來判斷患者的身體有無異常。輸液速度感測器將測得數據經 Arduino 計算點滴速度回傳至伺服器中，並即時顯示於網頁上。輸液速度控制器由護理師自網頁發送之點滴速度指令，依照其數值調整點滴速度，與輸液速度感測器共同組成「輸液監控器」。

2. 伺服器單元

使用 node.js 及 Python 建立了三套資料處理系統，分別是 Data server、Web server 和資料轉換模組



圖二：資料處理系統架構圖

i. Data Server

將 Arduino 之數據透過 mosquitto(MQTT Server)傳送，並將這些訊息存入 My SQL(Data Base)中。

ii. Web Server

處理由網頁送來的請求提供網頁的基礎架構，和發送由網頁對 Arduino 所下的指令，以及將 data base 中的資料推送至網頁上。

iii. 資料轉換模組與機器學習

讀取心電感測器數據後，由 Python 進行資料處理，將數據轉換為機器學習可用的資料型態再將資料送至機器學習演算法。使用訓練完成之機器學習模型基於回傳的心電圖資訊，判斷病人目前可能發生之病情後，發送提醒給護理人員，並微調輸液速度，保障患者身體情況。

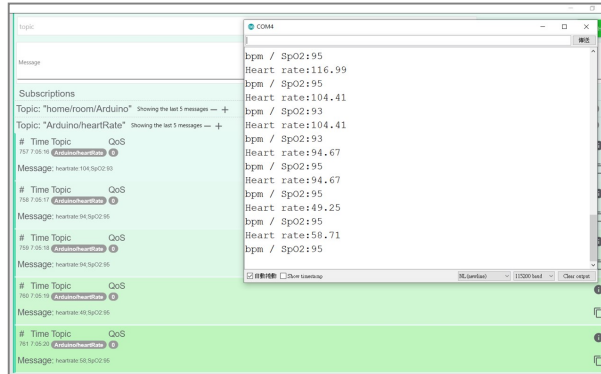
3. 使用者介面

此介面為建立給護理師使用的網頁，透過 Web Server 讀取 MySQL 資料庫所存取之數據，再將此數據即時更新至網頁中。網頁內容包含病患的身體數值、點滴速度、輸液結束提醒和能直接用網頁遠端控制輸液速度之功能。

(二) 實現方法

1. 監控硬體之生理感測

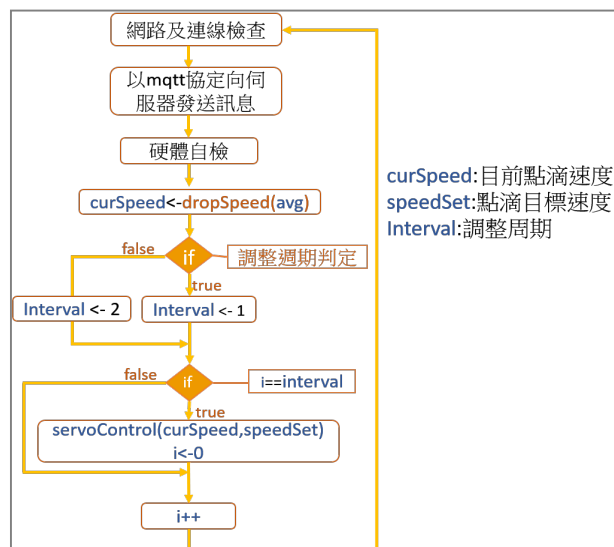
- i. 心率感測: 心率感測器目前用的是 MAX30100，將手指放在感測器上後即可讀取到脈搏與血氧，Arduino 會再將脈搏資料傳回電腦。圖三為將讀取到的心率與血氧資料同步傳給 MQTTLens。



圖三: 傳送心率資料

- ii. 心電感測: 在測量心電圖的部分，採用 AD8232，透過黏貼於身體上的電極貼片讀取心臟搏動的微弱電訊號，每隔約 25 毫秒採樣一筆數據。由於心電感測器的頻率較高，因此採用批次傳送的方式，每累計 60 筆資料再一次回傳至伺服器中，降低伺服器壓力。
- ## 2. 監控硬體之輸液監控

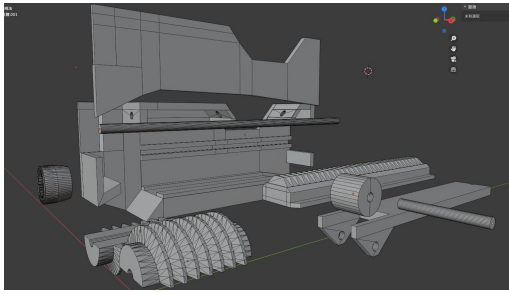
在輸液監控的部分，程式總體必須負責三個不同的工作，分別是維持網路連線、感測輸液速度以及控制點滴速度，圖四是主程式流程圖，在主程式運行前會有一個簡短的初始化程式，此初始化會將輸液速度感測器的背景平均感測值儲存為 avg 並於主程式中調用。



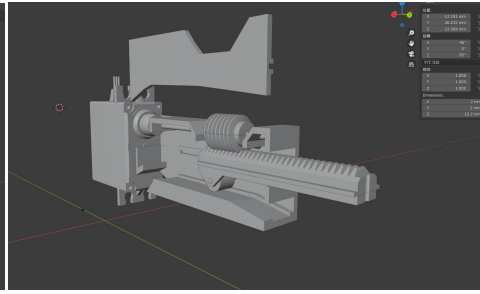
圖四: 輸液監控主程式流程圖

- i. 輸液速度控制器

輸液速度控制器採用 3D 列印的方式製作，藉由參考目前市面上常見的手動輸液速度控制器，設計出一款能遠端操作的輸液速度控制器，以擠壓外管的方式，配合 Arduino，使用伺服馬達控制即可遠端控制輸液的速度(如圖五)。



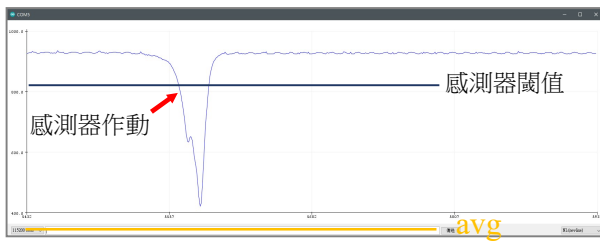
圖五(a): 零件分解



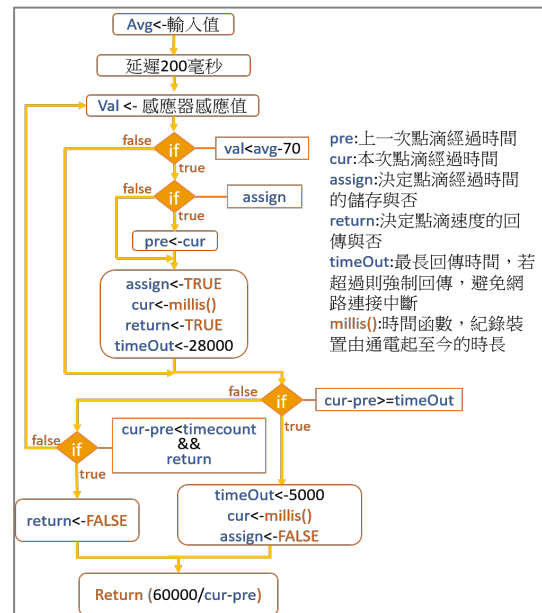
圖五(b): 組裝結構圖

ii. 輸液感測器

此感測器運用光遮斷開關的原理，使用一對紅外線收發管置於輸液套管滴室的兩側，在初始化程式中測得無液滴經過時的背景值 avg 後(圖六)，感測器便會在液滴經過時，測得一遠低於 avg 的感應值並以此作為判斷，紀錄兩滴液滴之間的時間差，便能精確地計算出當下輸液的速度，單位為滴／每分鐘。圖七是輸液感測器之實際運作流程。



圖六: 感測器原始訊號



圖七: 輸液感測器程式流程圖

iii. 輸液速度控制邏輯

為同時維持點滴調整至目標值的速度與調整時的穩定性，在點滴調整的部分引入以下控制邏輯，將每次液滴滴下時的調整量定義為 α ，令單位調整量 $\alpha = \varphi / T$ ，目前點滴速度 = v ，目前速度與目標的差 = Δ ，其中各項參數如下

$$T = \begin{cases} 1, & v \leq 12 \text{ or } \Delta \geq 15 \\ 2, & \text{else} \end{cases} \quad (1)$$

$$\varphi = \begin{cases} r * \Delta, & 3 < \Delta < 15 \\ k_1, & \Delta \leq 3 \\ k_2, & \Delta \geq 15 \end{cases} \quad (2)$$

$$r = \begin{cases} 30, & v \leq 7 \\ 20, & 7 < v \leq 25 \\ 10, & v > 25 \end{cases} \quad (3)$$

iv. 輸液結束提醒

此功能採用一個反射式紅外線感測器實作，藉由輸液瓶內有無液體時反射率的不同，當輸液即將結束，固定於輸液瓶身的感應器感應到數值不同，便會透過網頁向護理人員發出提醒。

3. 伺服器及網頁架設

由於 Arduino 對比 MySQL 為較高階的程式又因其較大的資料傳輸量，可能使資料異步接收，當同時有許多感測器回傳資料時，會需要排隊且造成封包丟失，所以本研究採用 Node.js 撰寫 Web Server 及 Data Server，用以整理資料使同步進行傳送。

i. Data Server

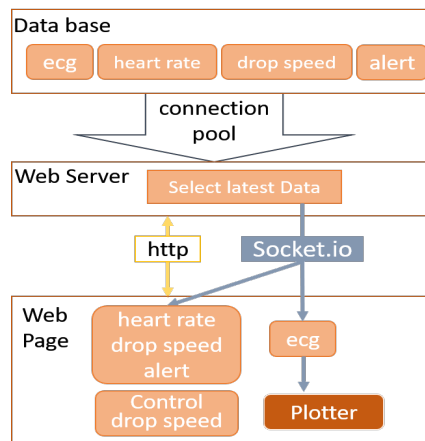
Data Server 會透過 MQTT server 承接 MQTT 協定下的感應器數據再分別存入 MySQL 資料庫的四個資料表中，資料表分別存取三個感測元件數據和通知訊息(表一)。

表一:MySQL 資料庫中資料

ecg	存取心電圖之數據
heartrate	存取心率之數據
dropspeed	存取點滴速度之數據
alert	存取裝置或生理狀況異常之訊息

ii. Web Server 和 Web page

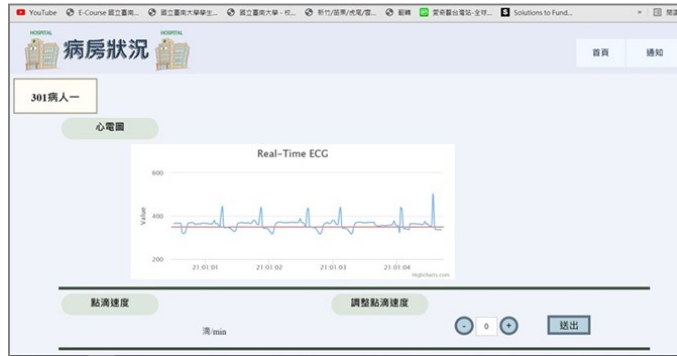
Web Server 利用連線池取得資料庫中最新一筆數據，再藉由 Socket.io 模組將訊息從伺服器推送至網頁中(圖八)，就能即時在網頁中顯示最新數據。



圖八:各模組間傳遞資料關係圖

· 心電圖繪製模組

由於資料庫中 ecg 資料表之數據為一長串的批次資料，因此在心電圖繪製模組中，會先將資料分割，再藉由 Highcharts 軟體庫繪製成動態心電圖如圖九。



圖九:心電圖

- 警示燈和通知設定

每個病患的視窗旁邊會有一個警示燈，當病患的輸液即將結束時，警示燈由綠色會轉變為黃燈；當發生硬體錯誤或生理數值異常時，警示燈轉變紅燈，並於開啟網頁時詢問護理人員是否開啟通知，如允許將在病人數值發生異常或點滴裝置連線不良時，會在電腦及手機通知欄中跳出通知提醒護理人員。

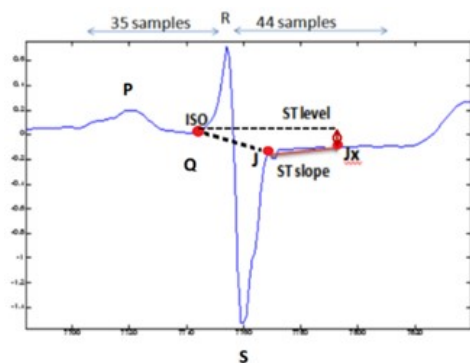
4. 機器學習

i. 資料讀取

要進行心肌缺血的判斷之前，由心電感測器搭配取得心電圖數據。

ii. 資料處理

為了能在心電圖中感測心肌缺血，必須取得 ST 區間特徵，因此要先取得較明顯的 R 點，再利用 R 點定義 SR 區間。我們將參考 Pan and Tompkin 提出的 R 點檢測演算法[6]。得到 R 點與 ST 區間後再取得其他與心肌缺血判讀相關的重要特徵如圖十，表二為這些特徵的定義。



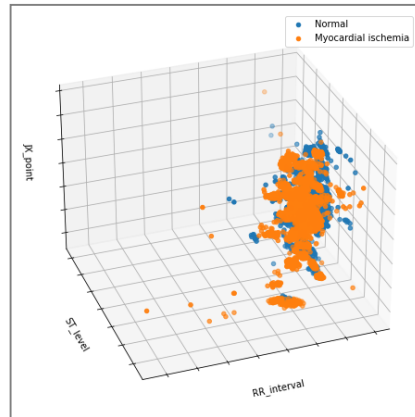
圖十:重要特徵

NO.	Feature	Feature Description
1	RR interval	計算 R_n 到 R_{n+1} 的時間長度
2	ST slope	計算J點到JX點之間的斜率
3	ST interval	當前J點到JX點之間的時間長度
4	ST level	計算ISO點與JX點的電位差
5	JX point	當前JX的點電位

表二:各特徵點定義

iii. 機器學習模型訓練

本研究對常用的機器學習模型隨機森林、決策數及 Knn 進行比較，取用 European ST-T Database[7]作為訓練數據集，將資料處理成 5 個特徵(圖十)與兩種類別(正常狀態與心肌缺血)，圖十一是正常狀態與心肌缺血在三個不同特徵(RR_interval、ST_level、JX_point)的分布圖，將訓練數據集丟入不同模型訓練、調整參數，最終進行模型評估，取得最適當的模型。

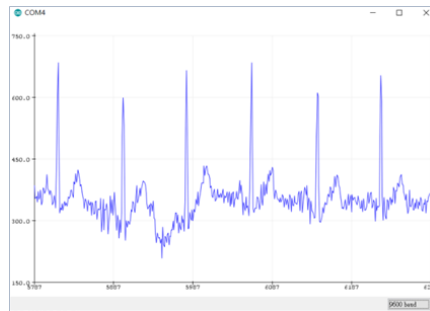


圖十一:特徵分布圖

4. 實驗結果

(一) 心電感測器

AD8232 是一款集成的信號調理模塊，得到的心電圖原始訊號如圖十二

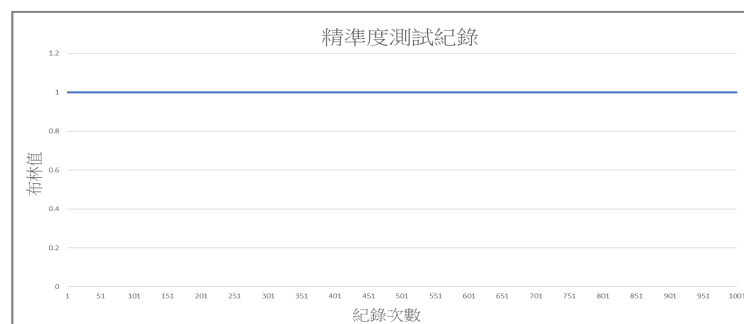


圖十二:心電圖原始訊號

(二) 輸液監控器

1. 輸液速度感測器

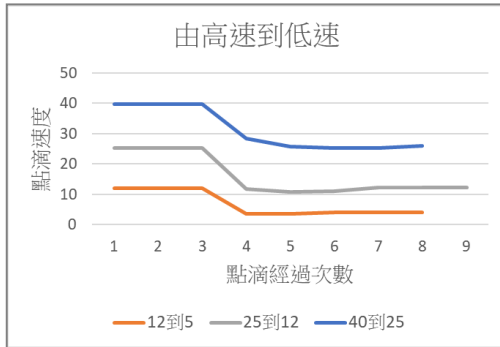
圖十三為輸液速度感測器測試紀錄，縱軸為紀錄值，2 代表液滴無落下但感測器有感應，1 代表液滴落下且感測器有感應，0 代表液滴落下但感測器無感應，以人工紀錄的 1000 次中並沒有發現任何誤作動的狀況發生。



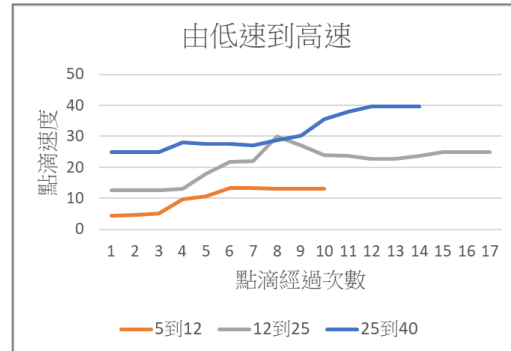
圖十三:精準度測試紀錄

2. 輸液速度控制器

輸液速度控制器的部分，分別測試了三個速域上調及下調的反應，由於調整週期是以液滴經過感測器為主，因此在圖十四中是以液滴經過次數作為橫軸而不是以時間作為橫軸。由結果可以發現下調時，皆能在 10 次液滴經過次數內達成，而上調時則是在 20 次內皆能達成。

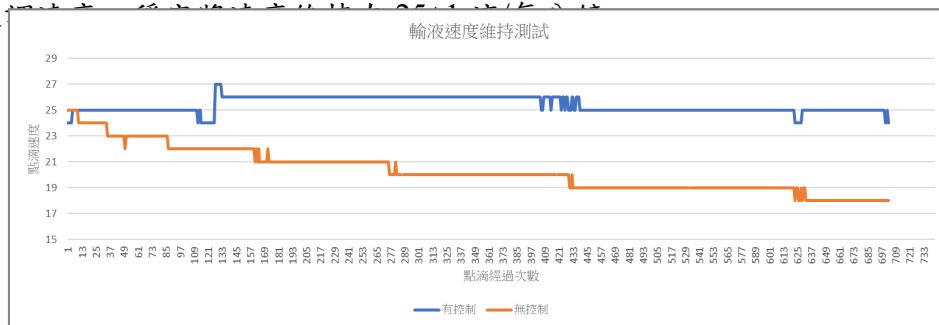


圖十四(a): 輸液速度下調



圖十四(b): 輸液速度上調

除了調整速度時的反應外，也針對速度控制器在維持速度的表現上，以 25 ± 1 滴/每分鐘為目標值做了測試。由圖十五中可觀察到，在無使用控制器的情況下，輸液速度會因為輸液瓶內液體減少而逐漸下降，而有使用控制器的情況下，只要速度低於目標值一段時間，控制器便會上

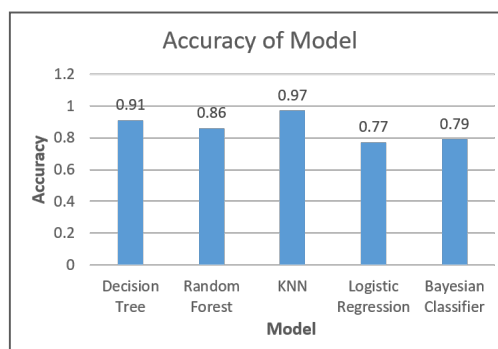


圖十五: 速度維持測試

(三) 機器學習

1. 演算法比較

本章節將使用 European ST-T Database 此資料庫[5]內的心電圖數據來進行演算法之間的比較，所使用到的常見演算法為 Random Forest、Decision Tree、KNN、Logistic Regression、Naïve Bayes，而演算法的驗證為 k-fold cross validation，由圖十六可看出 KNN 模型的準確度最高，且由圖十七去推算模型穩定度及特意度，也都為 KNN 獲得較好的成績，因此最終選擇 KNN 模型。



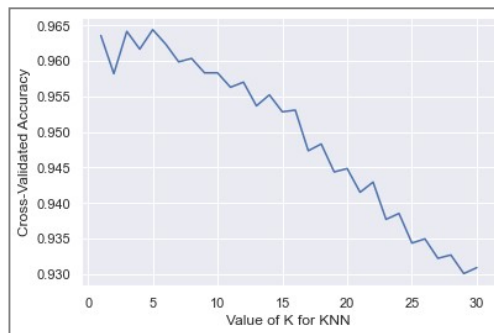
圖十六：不同模型之準確度比較

Decision Tree	Random Forest	Bayesian Classifier	KNN	Logistic Regression																																																												
<table border="1"> <tr> <td>predicted label 0</td> <td>256</td> <td>87</td> </tr> <tr> <td>predicted label 1</td> <td>326</td> <td>1851</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> </tr> <tr> <td></td> <td colspan="2">true label</td> </tr> </table>	predicted label 0	256	87	predicted label 1	326	1851		0	1		true label		<table border="1"> <tr> <td>predicted label 0</td> <td>343</td> <td>19</td> </tr> <tr> <td>predicted label 1</td> <td>56</td> <td>1262</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> </tr> <tr> <td></td> <td colspan="2">true label</td> </tr> </table>	predicted label 0	343	19	predicted label 1	56	1262		0	1		true label		<table border="1"> <tr> <td>predicted label 0</td> <td>179</td> <td>150</td> </tr> <tr> <td>predicted label 1</td> <td>388</td> <td>1803</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> </tr> <tr> <td></td> <td colspan="2">true label</td> </tr> </table>	predicted label 0	179	150	predicted label 1	388	1803		0	1		true label		<table border="1"> <tr> <td>predicted label 0</td> <td>355</td> <td>12</td> </tr> <tr> <td>predicted label 1</td> <td>37</td> <td>1276</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> </tr> <tr> <td></td> <td colspan="2">true label</td> </tr> </table>	predicted label 0	355	12	predicted label 1	37	1276		0	1		true label		<table border="1"> <tr> <td>predicted label 0</td> <td>271</td> <td>295</td> </tr> <tr> <td>predicted label 1</td> <td>1640</td> <td>6191</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> </tr> <tr> <td></td> <td colspan="2">true label</td> </tr> </table>	predicted label 0	271	295	predicted label 1	1640	6191		0	1		true label	
predicted label 0	256	87																																																														
predicted label 1	326	1851																																																														
	0	1																																																														
	true label																																																															
predicted label 0	343	19																																																														
predicted label 1	56	1262																																																														
	0	1																																																														
	true label																																																															
predicted label 0	179	150																																																														
predicted label 1	388	1803																																																														
	0	1																																																														
	true label																																																															
predicted label 0	355	12																																																														
predicted label 1	37	1276																																																														
	0	1																																																														
	true label																																																															
predicted label 0	271	295																																																														
predicted label 1	1640	6191																																																														
	0	1																																																														
	true label																																																															

圖十七：不同模型的混淆矩陣

2. 參數選擇

圖十八為不同 k 值下 KNN 模型的交叉驗證準確度，其最高準確度為 $k = 5$ ，而因樣本數： $8397 > 2^5$ (特徵數=5)且 kd_tree 的运算時間較短，因此 algorithm 選擇 kd_tree，將參數不斷進行比較，最終參數選擇為: n_neighbors=5、p=1(曼哈頓距離)、weights=uniform、algorithm=kd_tree。



圖十八:不同 k 值下的準確率

(四) 使用者操作介面

此介面為建立給護理師使用的網頁。網頁內容包含顯示病患的身體數值、點滴速度、輸液結束提醒、異常提醒和能直接用網頁遠端控制輸液速度之功能。

1. 數據監控介面

可透過 Web Server 讀取 MySQL 資料庫所存取之病患生理數據，再將此數據即時更新至網頁中，也可由網頁下達控制指令傳送至伺服器中，以遠端控制輸液速度。

2. 心電圖繪製模組

由於資料庫中 ecg 資料表之數據為一長串的批次資料，因此在心電圖繪製模組中，會先將資料分割，再藉由 Highcharts 軟體庫繪製成動態心電圖。

3. 警示燈和通知介面

每個病患的視窗旁邊會有一個警示燈，當病患的輸液即將結束時，警示燈由綠色會轉變為

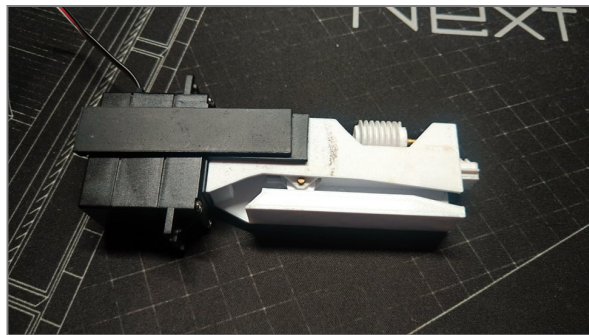
黃燈；當發生硬體錯誤或生理數值異常時，警示燈轉變紅燈，並於開啟網頁時詢問護理人員是否開啟通知，如果選擇允許，將會在病人數值發生異常或點滴裝置連線不良時，於電腦及手機通知欄中跳出通知提醒護理人員。

5. 實作成品展示

(一) 輸液監控器之成品



圖十九(a): 輸液感測器成品



圖十九(b): 輸液控制器成品

(二) 不同裝置之顯示介面

1. 手機版
2. 電腦版



圖二十(a): 電腦版



圖二十(b): 手機版

(三) 網頁功能展示

1. 即時視查病患生理數據及心電圖

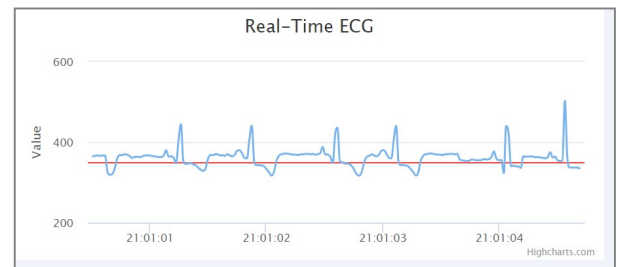
當病患在醫院或遠距醫療場所時，由於護理人員需一次照護多位病人，無法時時關注每一位病患之生理狀況，此系統將病患之生理數據實時顯示至網頁中，方便醫療人員在護理站電腦和手機上即時查看特定被照護者之生理數據。

- i. 護理人員可選取想查看之病房(圖二十一)

- ii. 查看即時顯示數據:隨著感測資料不斷更新，網頁中的數值也會隨之變動，不須重新整理頁面，網頁也會一直讀取最新資料顯示出來
- 顯示動態心電圖(圖二十二)
 - 顯示當下心律數據(圖二十三)
 - 顯示當下輸液速度(圖二十四)



圖二十一:選擇病房頁面



圖二十二: 顯示動態心電圖



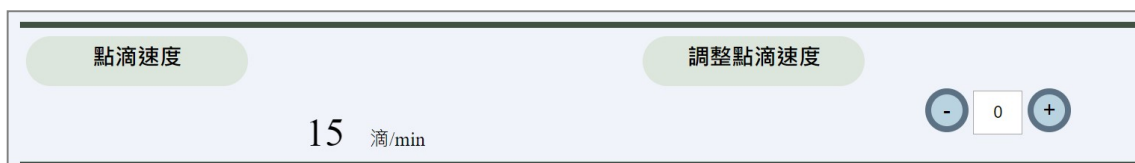
圖二十三: 顯示心律介面



圖二十四:顯示輸液速度介面

2. 智慧調整輸液速度

- i. 當病患在遠距醫療場所，如:照護中心、家中，醫院之護理人員可以使用此遠距醫療照護系統即時查看病患生理數據及心電圖，再根據專業判斷下達指令調整病人點滴速度。醫護人員可利用網頁輸入目標輸液速度，系統將會根據於使用介面下達的指令使護理人員不須親臨現場就能調整輸液速度。(圖二十五)
- ii. 機器學習模組由心電感測器感測之數據進行心肌缺血病症分析，將會自動判斷病情以調整輸液速度。



圖二十五:操作和顯示介面

3. 異常狀況緊急通知

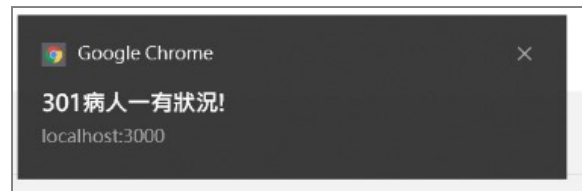
使護理人員不用時時查看病人之情況，當裝置異常或機器學習模型判斷病患有心肌缺血症狀時，發出提醒至手機及電腦告知護理人員。

i. 跳出通知:第一次進入網頁會跳出要求權限視窗(圖二十六)

· 點選允許:當系統裝置或病人有狀況時會在電腦及手機通知欄中跳出通知(圖二十七)



圖二十六: 要求權限視窗



圖二十七: 電腦通知欄跳出通知

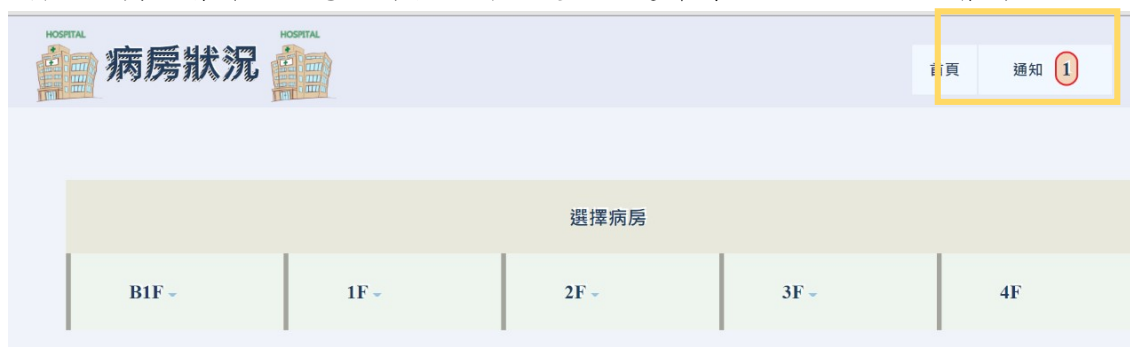
· 點選封鎖:通知僅會在開啟此網頁才會跑出通知(圖二十八)



圖二十八: 開啟網頁跳出通知

4. 照護人員遠距處理

系統保存尚未處理的通知於使用者介面，當護理人員看到網頁上方之通知欄位有通知(圖二十九)或特定病人右方之警示燈出現紅燈或黃燈(圖三十)可點選網頁進入通知頁面(圖三十一)提供醫護人員查看詳細訊息。當狀況排除後可點選清除按鈕，通知將會消失。



圖二十九: 通知欄位有一則通知

i. 病人警示燈

· 亮紅燈表示:機器學習判斷出病人生理有狀況或硬體裝置異常(圖三十 a)

· 亮黃燈表示:點滴即將用完(圖三十 b)

ii. :通知頁面顯示各種異常訊息 (圖三十一)

· 機器學習判斷出病患生理狀況異常:護理人員可調整點滴速度或前去查看病人狀況

· 點滴即將用完:護理人員即可拔掉點滴

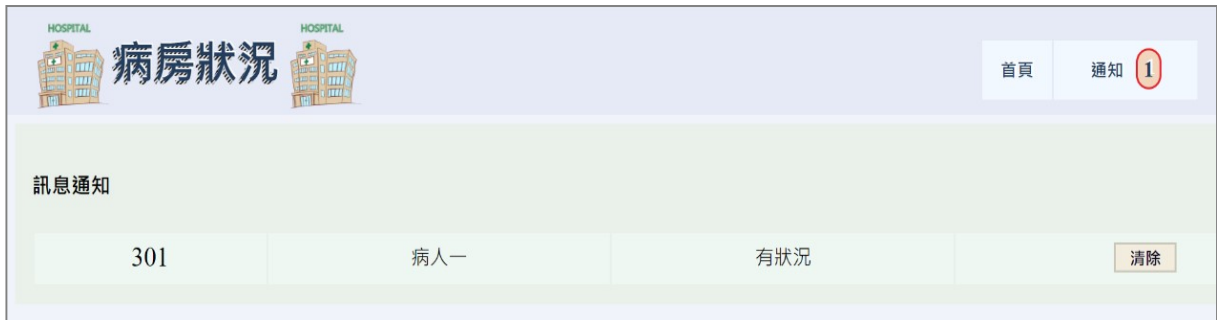
· 硬體異常:可進行檢查動作



圖三十(a):亮紅色警示燈



圖三十(b):亮黃色警示燈



圖三十一(a):通知介面顯示



圖三十一(b):通知介面顯示

6. 結論

本團隊最終的目標是大範圍的應用於每個有需求的醫療院所，並且將所有能以心電圖判讀出的疾病加入本系統機器學習的辨識範圍。本次研究的成果已經成功地展現出此系統的可能性，不論是在輸液系統的遠端操控、即時的生理數據顯示抑或是機器學習對疾病的判讀，皆已到達本團隊對階段性目標的預期。未來，在系統發展更加成熟後，便能更進一步增加對病情的分析，達到能預測患者未來病情走向的功能，輔助醫療以及長照環境的進步。

7. 致謝

這項研究是由科技部大專學生研究計畫 MOST10-2813-C-024-004-E, MOST107-2221-E-024-001-MY3, MOST 109-2221-E-024-013 所補助。

8. 參考文獻

文獻之編號按出現之先後次序編列，範例如下：

- [1] Doherty, M., & Buggy, D. J. (2012). Intraoperative fluids: how much is too much?. *British journal of anaesthesia*, 109(1), 69-79.
- [2] Mitchell, T. M. (1997). Does Machine Learning Really Work?. *AI Magazine*, 18(3), 11. <https://doi.org/10.1609/aimag.v18i3.1303>
- [3] Arsanjani, R., Xu, Y., Dey, D. et al(2013). Improved accuracy of myocardial perfusion SPECT for detection of coronary artery disease by machine learning in a large population. *J. Nucl. Cardiol.* 20, 553–562. <https://doi.org/10.1007/s12350-013-9706-2>
- [4] MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press. ISBN 0-521-64298-1.
- [5] Altman, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*. 1992, 46 (3):175-185. [doi:10.1080/00031305.1992.10475879](https://doi.org/10.1080/00031305.1992.10475879)
- [6] J. Pan and W. J. Tompkins, “A Real-Time QRS Detection Algorithm,” *IEEE Transactions On Biomedical Engineering*, vol. 32, pp. 230-236, 1985
- [7] Taddei A, Distanti G, Emdin M, Pisani P, Moody GB, Zeelenberg C, Marchesi C, “The European ST-T database: standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography,” 1992 *The European Society of Cardiology*, pp. 1164-1172, 1992

