

在室內環境中基於機器學習與物聯網的智慧節能控制系統

An Intelligent Energy-Saving Control System Based on Machine Learning and Internet of Things in Indoor Environment

王能中 蔡宛昀 方昱晴 張瑋哲

國立聯合大學資訊工程學系

苗栗市南勢里聯大二號

ncwang@nuu.edu.tw

{U1118016, U1124004, U1124022}@o365.nuu.edu.tw

摘要

本研究旨在建置一套應用於室內環境的智慧節能控制系統，結合物聯網 (IoT)、機器學習 (ML)、環境感測與紅外遙控技術，以提升能源效率並維持熱舒適度。系統採三層架構，包括物聯網層、後端服務層與前端介面層。物聯網層整合 LifeSmart 溫溼度感測器、智慧插座、人體感測器與飛利浦 WiZ 智慧燈泡，提供環境與用電資料。後端以 Node.js 建置伺服器，並使用 CatBoost 模型預測最適風扇風速與冷氣調整量，搭配 Tuya IoT 平台執行紅外線控制。前端以 Flutter 打造跨平台介面，提供環境監控、裝置操作、用電視覺化與異常通知等功能。

研究以 PMV (Predicted Mean Vote) 作為熱舒適度指標，藉由感測數據與模型推論達成冷氣與風扇的自動化節能控制。系統具備監測、節能、自動化、手動控制、即時通知與歷史紀錄等功能，成功建置一套可應用於教室、宿舍與辦公室等室內場域的智慧節能平台，有效降低能源消耗並改善環境品質。

關鍵字：梯度提升樹、物聯網、機器學習、智慧節能、熱舒適度。

Abstract

The objective of this study is to develop an indoor smart energy-saving control system that integrates Internet of Things (IoT) devices, machine learning models, and environmental sensing technologies. The system adopts a three-layer architecture consisting of the IoT layer, backend application service layer, and frontend interface layer. Various devices—including temperature and humidity sensors, smart plugs, motion detectors, and WiZ smart lighting—are used to collect environmental and power consumption data. The backend, built with Node.js, employs a CatBoost regression model to predict optimal fan speed and air-conditioner temperature adjustments, while control signals are transmitted through the Tuya IoT platform for infrared-based device operation.

The system incorporates PMV (Predicted Mean Vote) as the thermal comfort evaluation index,

enabling environment-aware automated cooling control. A cross-platform user interface developed with Flutter provides real-time monitoring, device control, energy visualization, notifications, and basic system configuration. The system demonstrates its ability to reduce energy consumption while maintaining indoor comfort, making it applicable to semi-public indoor spaces such as classrooms, dormitories, and offices. This work establishes a compact and extensible smart energy-saving platform with practical potential for smart-building and smart-campus environments.

Keywords: Gradient Boosting Tree (CatBoost), Internet of Things (IoT), Machine Learning, Smart Energy Saving, Thermal Comfort.

1. 簡介

1.1. 前言

隨著全球暖化問題日益加劇，節能減碳與環境永續是刻不容緩的關鍵議題，如何有效利用科技實現智慧化能源管理，成為現代建築與生活空間的挑戰。傳統電器設備多以固定模式運行，無法即時因應環境變化而調整，使得能源使用效率低落，造成不必要的電力浪費。尤其在學校教室、學生宿舍、辦公室等半公共室內場域，能源效率不僅直接影響營運成本，更關乎使用者體驗與碳足跡表現。

然而，隨著物聯網 (IoT) 與機器學習 (ML) [1][2] 技術的快速發展，使得即時感測、資料分析與自動化控制得以更有效率地實現。低功耗環境感測器及智慧插座等設備日漸普及且價格親民，大幅降低了智慧節能系統的建置門檻，加速其在智慧建築與智慧校園中的應用。

有鑑於此，本研究旨在整合環境感測器、機器學習模型與用電量監控設備，設計並建構一套能夠即時判斷室內環境狀態、自動調整冷氣與風扇運行參數以及控制設備開關的智慧節能控制系統。期望藉此有效提升能源使用效率，並同時兼顧使用者的舒適感。

1.2. 研究動機

隨著都市建築與能源消耗增加，且人們對室內舒適度的需求越來越高，傳統的空調與燈光控制多依賴固定運行或手動調節，既耗能又不靈活。市面上雖已有智慧家居與節能控制系統，為使平台介面得以依研究需求客製化並避免受商用系統限制，因此選擇自行建置。自建平台讓我們能自由整合各種感測器、調整控制策略，並驗證 AI 模型在實際環境下的效果。

1.3. 相關研究

Gubbi 等人所提出的物聯網三層架構，為智慧節能系統的平台建構提供了清晰的理論基礎[3]。在此架構下，Deb 等人歸納並證明了多元回歸、時間序列與深度神經網路等機器學習模型，能有效應用於空調負載與用電量的預測。同時，Fayyad 等人提出的從資料中發掘有用知識 (KDD) 處理流程，為本研究的資料前處理階段提供了系統化的方法論。

本研究以 Fanger 提出的預測平均投票 (Predicted Mean Vote, PMV) 指標模型為核心，作為評估人體熱舒適度[4][5]的重要依據。此外，Charles 與 Lin 等人針對亞熱帶地區的研究，提供了具體的實測數據，指出不同性別感到舒適的空調溫度範圍 (23.4~25.8°C)。Wong 等人的研究進一步證實，不當的溫度設定不僅會影響使用者的舒適體驗，更會造成不必要的能源浪費。

近年的許多研究致力於透過粒子群演算法 (Particle Swarm Optimization, PSO)、模糊控制與增強式學習等智能演算法，動態調控空調與風扇參數，以達到最理想的節能與舒適度平衡。

2. 系統內容

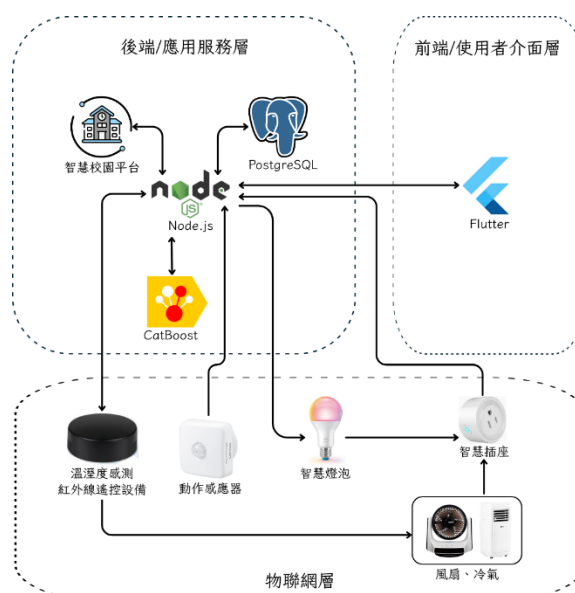
2.1. 系統架構

本智慧節能控制系統的設計採用三層架構，確保數據流的順暢與控制指令的即時性。此架構包含物聯網層、前端 (使用者介面層)、後端 (應用服務層)。

在物聯網層中，透過 LifeSmart 溫溼度感測器紅外線遙控設備讀取溫溼度並控制家電 (如風扇或冷氣)。同時，系統也可以透過 Wi-Fi UDP 通訊控制飛利浦 WiZ 智慧燈泡，以及使用智慧插座監測家電用電量，實現多裝置間的智慧化整合與能源管理。

這些資料會透過網路傳送至後端 (應用服務層)，由 Node.js 伺服器進行資料處理與邏輯運算，並將溫溼度及用電量數據存入 PostgreSQL 資料庫。此外，後端也整合非線性梯度提升回歸模型，用以分析環境變數並預測最佳節能設定。後端系統能與智慧校園平台連線，實現資料視覺化。

在前端 (使用者介面層)，以 Flutter 建立跨平台應用，提供即時數據視覺化與裝置控制介面，使用者可透過手機或電腦即時監控室內環境並調整家電設定，達到節能與舒適的平衡，如圖一所示。



圖一：系統架構圖。

2.2. 開發環境

本系統的開發環境，說明如下。

A. Visual Studio Code

Visual Studio Code 是一種由 Microsoft 開發的免費開源程式碼編輯器。開發者可透過豐富的擴充元件庫客製化工作環境，以提升開發效率。本系統中，我們選用 Visual Studio Code 作為整個系統的主要開發環境。

B. Python

Python 是一種高階、通用且直譯式的程式語言。本系統中，我們選用 Python 撰寫資料分析與機器學習模型、資料處理流程。

C. JavaScript

JavaScript 是依據 ECMAScript 標準的直譯式程式語言，具備事件導向與非阻塞式特性，常用於建置動態網頁互動功能。本系統之後端開發採用 Node.js，以 JavaScript 作為主要語言，並透過大量套件生態系 (npm) 支援如資料存取、API 建置與排程等功能，使系統開發流程更具效率與擴充性。

D. Dart

Dart 是由 Google 開發的物件導向程式語言，具備強型別、安全性與高效能等特性，易於學習並適合大型應用程式開發。Dart 最大特色在於可透過 Flutter 框架進行跨平台介面開發，將單一程式碼編譯至 Android、iOS、Web 與桌面平台。本研究使用 Dart 作為前端應用的主要語言，以整合 Flutter 的高度可擴充 UI 組件與即時互動能力。

E. Node.js

Node.js 是一個開源、跨平台的 JavaScript 運行環境，它使用 Google Chrome 的 V8 JavaScript 引擎來高效執行程式碼。Node.js 使得 JavaScript 不再僅限於瀏覽器端，而能夠在後端、物聯網設備

以及各種應用程式中運行。本系統中，選用 Node.js 作為網頁後端的主要開發環境。

F. Flutter

Flutter 是一個由 Google 開發的開源 UI 軟體開發工具包，用於從單一程式碼庫構建美觀、原生編譯的 app、網頁和桌面應用程式。它以 Dart 語言為基礎，旨在提供開發者一個快速、高效且富有表現力的 UI 開發體驗。本系統中，選用 Flutter 作為資料視覺化與前端界面的主要開發環境。

G. PostgreSQL

PostgreSQL 是一款開源、跨平台且高度可靠的關聯式資料庫管理系統。其支援標準 SQL 語法，並提供強大的進階功能，例如複雜查詢、索引優化、觸發器、儲存程序與多版本並行控制。

在本系統中，PostgreSQL 用於整合環境感測、設備控制、用電紀錄、警示與使用者行為資料。良好的資料一致性與查詢效率，使系統能可靠地進行長期資料蒐集與後端服務運算，並支援後續資料分析與模型推論流程。

2.3. 相關技術

本系統的相關技術，說明如下。

2.3.1. PMV

預測平均投票 (Predicted Mean Vote, PMV) 指標為丹麥學者 P.O. Fanger 教授於 1972 年所發表人體熱平衡模型，該模型用來表示人體對於環境中冷、熱的感受，並以預測不滿意百分率 (Predicted Percentage of Dissatisfied, PPD)，表示在該 PMV 舒適指標中，有多少百分比的人感到不舒適。其模型同時考量六項主要熱舒適因子：空氣溫度、相對濕度、氣流速度、平均輻射溫度、代謝率 (met) 與服裝隔熱值 (clo)，並根據如圖二、圖三所示的公式，計算出 PMV 指標值與 PPD 指標值，對應至範圍-3(非常冷)至+3(非常熱)的主觀感受尺度。PMV 能將客觀環境條件轉換為可量化的人體舒適度指標，因此廣泛應用於室內環境品質評估、HVAC 優化控制，以及智慧建築之能源管理策略設計。本系統使用 PMV 作為模型輸入，用以反映使用者的即時熱舒適狀態。

$$\begin{aligned} \bullet PMV &= (0.303e^{0.0036M} + 0.028) \times \{M - 3.05 \times 10^{-5} \times (5733 - 6.99M - P_a) - 0.42 \times (M - 58.15) - 1.7 \times 10^{-5} \times M \times (5867 - P_a) - 0.0014 \times M \times (34 - t_a) - 3.96 \times 10^{-8} f_{cl} \times [(t_{cl} + 273)^4 - (t_r + 273)^4] - I_{cl} \times h_c \times (t_{cl} - t_a)\} \\ \bullet t_{cl} &= 35.7 - 0.028 \times M - I_{cl} \times \{3.96 \times 10^{-8} f_{cl} \times [(t_{cl} + 273)^4 - (t_r + 273)^4] + f_{cl} \times h_c \times (t_{cl} - t_a)\} \\ \bullet h_c &= \begin{cases} 2.38 \times (t_{cl} - t_a)^{0.25} & \text{for } 2.38 \times (t_{cl} - t_a)^{0.25} > 12.1 (v_{ar})^{0.5} \\ 12.1 (v_{ar})^{0.5} \times (v_{ar}) & \text{for } 2.38 \times (t_{cl} - t_a)^{0.25} < 12.1 (v_{ar})^{0.5} \end{cases} \\ \bullet f_{cl} &= \begin{cases} 1.00 + 1.29 I_{cl} & \text{for } I_{cl} \leq 0.078 m^2 k / w \\ 1.05 + 0.645 I_{cl} & \text{for } I_{cl} > 0.078 m^2 k / w \end{cases} \\ \bullet P_a &= (RH / 100 \times e^{(18.6686 - 4030.18 / (TA + 235))}) / 0.00750062 \\ \bullet PPD &= 100 - 95 \cdot \exp(0.03353) \cdot PMV^4 - 0.2179 \cdot PMV^2 \end{aligned}$$

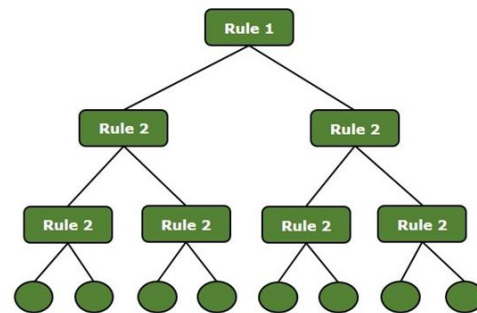
圖二：PMV、PPD 公式。

- M：人體新陳代謝量 (W/m²)
- W：外部影響人體之能量 (W/m²)
- I_{cl}：衣著隔絕度 (m²K/W)
- f_{cl}：衣著面積因素
- t_a：氣溫 (°C)
- t_r：平均輻射溫度 (°C)
- v_{ar}：相對風速 (m/s)
- P_a：水氣分壓 (Pa)
- h_c：對流熱傳係數 (W/(m²K))
- t_{cl}：衣著表面溫度 (°C)

圖三：PMV、PPD 公式。

2.3.2. CatBoost

CatBoost 屬於梯度提升決策樹 (Gradient Boosting Decision Trees, GBDT) 模型，其核心由多棵弱學習器所組成，而每一個弱學習器即為一棵決策樹。決策樹透過逐層切分特徵來最小化目標函數，使資料在葉節點產生較一致的預測值。CatBoost 以「序列式訓練」方式疊加多棵決策樹：每一棵樹負責修正前一棵樹的誤差，最終形成一個高準確度的集成模型，如圖四所示。



CatBoost Decision Tree

圖四：決策樹。

為避免梯度提升常見的目標洩漏，CatBoost 採用 Ordered Boosting，以隨機排列並「僅使用先前資料」的方式建立每一棵樹的統計特徵，提高模型的泛化能力。此外，CatBoost 對類別特徵採用內建編碼方式，無需人工進行 One-hot 等前處理，使決策樹能更有效地學習類別分布。綜合來看，CatBoost 透過多棵決策樹的梯度疊代、順序增強機制與類別特徵優化，使其在回歸與分類任務中具備高效能與穩定預測能力。

本系統採用 CatBoost 迴歸模型預測兩項控制參數：風速設定與冷氣溫度調整量。模型輸入特徵包含環境與人體舒適度相關變數，先將資料切分為訓練集與驗證集，再以 CatBoost 進行兩個獨立模型的訓練，並使用 RMSE 作為損失函數以提升預測精度。訓練完成後，兩個模型皆匯出為 ONNX 格式，提供後端預測服務快速載入與執行，實現即時的風扇風速與冷氣調節決策系統。

2.3.3. WiZ UDP 通訊機制

WiZ 智慧燈泡採用 UDP (User Datagram Protocol) 作為本地網路的主要控制通訊協定。其通訊流程以「無連線式、低延遲」為特性，控制端可直接在區域網路內透過指定的 UDP 埠向燈泡廣播或單點傳送 JSON 格式的控制指令。WiZ 裝置收到封包後會立即回應當前狀態或執行相應動作，如開關、調光或模式切換。由於 UDP 封包開銷低、傳輸速度快，適合即時性要求高的 IoT 控制場景，因此 WiZ 系統在不依賴雲端的情況下即可達成本地化、快速且穩定的裝置控制。本研究利用 WiZ 的 UDP 協定完成燈泡狀態讀取與控制功能，以提升系統反應時間與本地操作可靠性。

2.3.4. Tuya IoT Platform

Tuya IoT Platform (塗鴉智能物聯網平台) 是一個全球化的雲端物聯網開發平台，提供從硬體連接、雲端管理、App 開發到資料分析的完整解決方案。開發者可透過 Tuya 提供的 SDK、API 及雲端服務，快速實現智慧裝置的開發與管理，並將不同裝置整合於同一平台中進行監控與控制。

本研究透過 Tuya IoT Platform 所提供的雲端 API，實作智慧家電的控制與用電量監測功能。系統利用 Tuya 的紅外線控制 API 對空調設備發送 IR 指令，以達成遠端開關與模式調整；同時亦透過智慧插座的雲端資料介面取得即時用電量資訊，包含電壓、電流、功率與累積用電量等參數。

2.3.5. 物聯網智慧校園數據分析服務平台

該平台提供完善的物聯網數據管理功能。使用者登入後可在首頁查看各感測場域的位置與設備狀態，方便即時監控。平台支援三十多種感測器類型，且可自訂場域與感測器群組，並查閱即時與歷史數據。控制端可透過 MQTT、LoRa、NB-IoT 等技術進行裝置操作，亦支援邏輯及排程控制，並記錄開關歷程。(網址：<https://edu.nthu-smart-farming.kits.tw/>)

2.3.6. MQTT

MQTT (Message Queuing Telemetry Transport) 是一種輕量化、基於發布、訂閱模式的訊息傳輸協定，專為低頻寬、低功耗與不穩定網路環境設計。其通訊架構包含 Broker 與 Client 兩個角色：Client 可將訊息發布至特定主題，也可訂閱所需主題以接收即時訊息；Broker 則負責管理訊息路由與分發，確保發布的訊息安全送達訂閱者。由於訊息格式簡單、網路開銷小，MQTT 廣泛應用於物聯網、智慧家庭與感測器資料傳輸等場景，特別適合即時事件監測與控制系統。

2.3.7. Home Assistant

Home Assistant 為一套開源智慧家庭管理平台，具備高度擴充性與多協定整合能力。在本研究中，Home Assistant 透過 MQTT 通訊協定與人體動作感應器進行連接。MQTT 採用主題訂閱與發布模式，感應器會將偵測到的動作事件以訊息形式

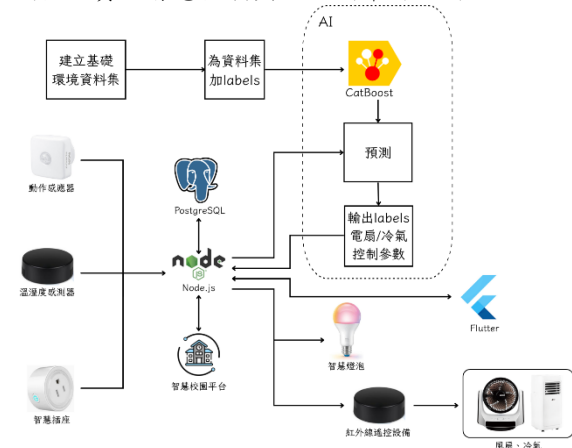
發布至指定主題；Home Assistant 則作為 MQTT client 訂閱相關主題並即時接收事件更新。此方式具備低延遲、低網路負載與高穩定性，使系統能在偵測人體活動後立即觸發後續自動化流程，例如照明控制、環境調節或行為紀錄。透過 MQTT 的輕量化架構，Home Assistant 能以通用方式整合不同廠牌與通訊協定的感測器設備，提升整體系統的互通性與擴充性。

2.4. 系統運作流程

本智慧節能控制系統的運作流程如圖五所示。系統首先由 LifeSmart 溫溼度感測器與智慧插座蒐集室內環境與用電數據，並透過人體動作感測器偵測是否有人。感測資料經網路傳送至後端 Node.js 伺服器，儲存於 PostgreSQL 資料庫及智慧校園平台，作為分析與模型訓練依據。

在模型訓練階段，系統建立環境資料集並加上標籤，使用 Python 執行 CatBoost 訓練後匯出模型供推論使用。推論時，後端根據即時感測數據預測風扇風速與冷氣調整量，並透過 Tuya IoT API 將控制指令傳至本地紅外線模組以執行設備調節。

同時，系統也會將設備狀態與環境資訊回傳至前端 Flutter 應用，使用者可進行即時監控與手動調整，實現智慧控制與人工調節的互補。



圖五：系統運作流程圖。

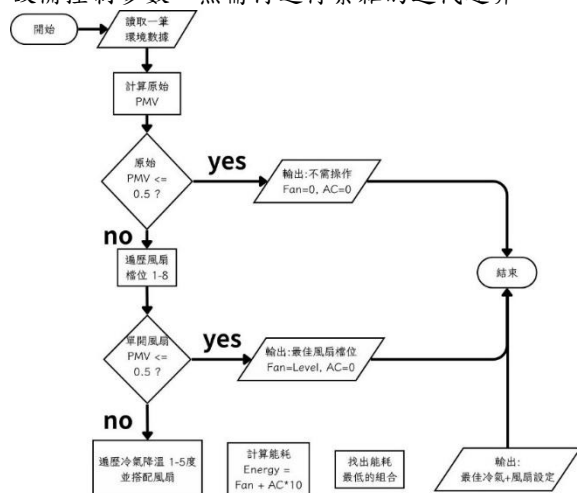
2.5. 模型流程

環境模擬資料集透過程式大規模的建立，依據 ISO 7730 標準中的 Fanger PMV 公式，針對溫度 (24-30°C)、濕度 (30-80%)、風速、代謝率及衣著量等參數進行全域遍歷，生成涵蓋各種室內情境的基礎熱舒適度數據。此階段確保了模型訓練資料的多樣性與物理正確性。

原始的環境數據僅反映了當下的舒適狀態，並未包含控制建議。因此，我們要對每一筆數據進行邏輯優化處理，為其加上標籤 (Labels)，流程如圖六所示。程式會模擬開啟不同檔位的風扇 (Level 1-8) 與冷氣降溫幅度 (1-5°C)，並計算風扇及電扇的各種組合後的 PMV 值。系統採用「最小能耗優先」的演算法，在確保 PMV 降至舒適區間的前提下，優先選擇風扇控制；若風扇不足以達到舒適，才介入冷氣控制，並尋找能耗最低的組合。

最終，每一筆環境數據都被標註了「最佳風扇檔位」與「最佳冷氣設定值」，作為監督式學習的標準答案。

模型訓練利用標註好的資料集輸入至模型進行監督式學習。模型學習環境參數（輸入特徵）與最佳控制策略（輸出標籤）之間的非線性關係。經過訓練後的模型被轉換為輕量化的 ONNX 格式，能夠部署於邊緣運算裝置。在實際應用時，系統僅需讀取感測器數值，AI 模型即可預測出最節能的設備控制參數，無需再進行繁雜的迭代運算。



圖六：資料集建立流程圖。

3. 系統功能

3.1. 感測資料與能耗監測系統

提供即時查詢功能，讓使用者能透過前端介面快速掌握最新感測資料。同時，透過歷史紀錄，完整保存所有過往能耗紀錄，為長期的趨勢分析提供穩固基礎。

3.1.1. 節能系統

節能系統負責根據環境狀態與模型預測的結果，自動調節家電設備，以達成舒適與節能的目標。系統首先根據環境感測數據與人體舒適相關變數計算舒適指標，並透過模型評估使用者的熱舒適度，再透過 CatBoost 回歸模型分別預測最佳風扇風速與冷氣溫度調整量，作為最終控制決策。當系統決定調節設備時，會使用 Tuya IoT Platform API 將紅外線控制指令傳送至雲端，並由本地 IR 模組發送至冷氣與風扇，完成開關機、溫度及風速調整等操作。透過 AI 模型、人體動作感測及環境感測數據的整合，本節能系統能自動化地維持室內舒適度，同時有效降低能源消耗，實現智慧化的環境控制。

3.1.2. 控制系統

本控制系統除了自動化節能與舒適度調節外，亦提供遠端控制功能，使用者可透過行動裝置或網頁介面即時操作家電設備。為實現此功能，我們對風扇與冷氣的遙控器按鍵進行學習，將每個操作指

令記錄並整合至系統中，讓使用者即使身處外地，也能透過網路遠端開關設備、調整風速或冷氣溫度，避免忘記關閉家電所造成的能源浪費與安全疑慮。此設計使控制系統兼具智慧自動化與遠端操控能力，提升使用便利性與節能效益。

4. 系統實作成果

本系統的實作成果包含：「使用者介面」、「遠端手動控制系統」及「自動節能控制系統」，說明如下。

4.1. 使用者介面

使用者介面提供了「網站」及「Android Application」，皆包含「使用者註冊、登入」、「首頁」、「燈光控制」、「風扇控制」、「冷氣控制」、「用電監控」、「省電效能展示」、「節能設定」、「我的帳戶」、「通知設定」、「通知歷史紀錄」及「修改帳戶資料」。說明如下。

4.1.1. 系統首頁的註冊登入

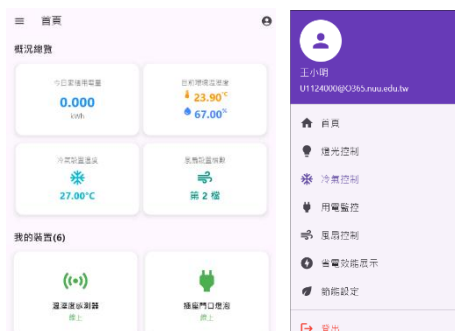
使用者可以透過電子郵件進行帳號註冊與登入。

4.1.2. 網站首頁

首頁呈現系統的整體概況，包含「今日累積用電量」、「目前環境溫溼度」、「冷氣設定溫度」及「風扇設定檔位」等關鍵資訊，讓使用者能快速掌握室內環境與設備運作狀態。首頁下方亦顯示所有已成功連線的裝置，協助使用者即時了解系統是否正常運作。此外，各功能頁面右上角均設置人像圖示，點擊後可進入「我的帳戶」頁面，如圖七所示。

4.1.3. 側邊欄

在各功能分頁的左上角設有側邊欄圖標，使用者可透過該選單於不同功能頁面之間切換；側邊欄底部則配置「登出」按鈕，讓使用者能在操作結束時方便地退出系統，如圖八所示。



圖七：首頁。 圖八：側邊欄。

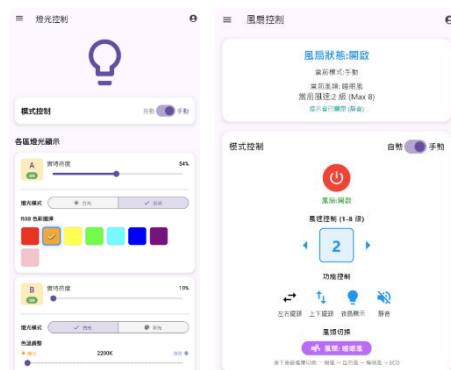
4.1.4. 燈光控制

本系統提供燈光控制介面，使用者可在「自動」與「手動」兩種模式之間切換。在自動模式下，系統會依據室內是否有人自動判斷並控制燈光開關；手動模式則讓使用者能自行調整燈光設定。介面中會顯示各區域的燈光控制資訊，例如區域 A 提供即時亮度顯示及 RGB 色彩調整功能，區域 B 則

提供亮度資訊與色溫調整,方便使用者依需求調整亮度、顏色與色溫。介面下方設置「燈光情境」區域,內含多種預設情境,如「日常情境」、「聖誕節」、「派對」與「萬聖節」等,讓使用者能快速套用合適的燈光效果,如圖九所示。

4.1.5. 風扇控制

本系統提供風扇控制介面,介面上方會顯示「風扇狀態」、「當前模式」、「當前風類」及「當前風速」等即時資訊。使用者可在下方切換「自動」或「手動」兩種操作模式;於自動模式下,風扇的開關與風速將由系統透過模型自動調節,以達到節能運作;而在手動模式中,使用者則可自行調整風速等級並控制風扇的開關。介面同時提供左右擺頭、上下擺頭、液晶顯示與靜音等實用功能按鈕,方便依需求進行設定。畫面底部則可選擇不同的風類模式,包括「一般風」、「自然風」、「睡眠風」與「ECO 溫控」等選項;其中 ECO 模式會根據室內溫度自動調整風速,如圖十所示。



圖九：燈光控制。圖十：風扇控制。

4.1.6. 冷氣控制

冷氣控制介面的頂部會顯示「當前環境資訊」,包含室內的溫度與濕度。使用者可在「自動」與「手動」兩種模式間切換;在自動模式下,冷氣的開關與設定溫度會由系統透過機器學習模型進行節能調控,而在手動模式中,則由使用者自行操控,可透過溫度控制區調整設定溫度並開啟或關閉冷氣電源(其中溫度調整僅在冷氣模式下可使用)。在運轉模式區塊中,介面提供「送風」、「自動」、「冷氣」與「除濕」等功能切換。此外,使用者亦可設定風速,包含低速與高速兩種,並可在冷氣模式下啟用睡眠模式,如圖十一所示。

4.1.7. 用電監控

用電監控介面讓使用者可透過頂部導航列快速切換檢視 1 號至 4 號各插座的使用資訊。即時資料區塊會清楚顯示當前選定插座的功率、電壓與電流,並在右上角呈現「開/關」狀態。使用者亦可在選擇日期後,以「日」、「週」、「月」等不同維度檢視歷史用電趨勢,透過趨勢圖與數據表格直觀掌握區間用電變化情況。此外,系統支援將資料匯出為 CSV 或 Excel 格式報表,方便後續分析與保存,如圖十二所示。



圖十一：冷氣控制。圖十二：用電監控。

4.1.8. 省電效能展示

省電效能展示介面以模擬比較與實際測試兩大面向呈現智慧節能的核成。使用者可在同一介面中對比傳統模式、風扇輔助與智慧節能模式的耗電差異,並透過圖表清楚看到智慧模式因自動控制與離席關閉所帶來的約 40 - 50% 節能效果。頁面同時提供 10 分鐘實測的真實功率變化,以及依據實測推估的每日、每月與每年節電金額,搭配即時環境監測與設備功率資訊,讓節能效果一目了然、可驗證且可量化,如圖十三所示。

4.1.9. 節能設定

節能設定介面讓使用者可依自身狀況調整影響 PMV 舒適度計算的參數。頁面上方提供設定說明,其下方顯示當前的環境資訊,包括溫度、濕度及模型建議溫度,並以指標圖呈現 PMV 舒適度狀態。環境資訊下方包含活動類型、穿著類型與空氣流速三項設定,使用者可點擊展開進行調整。頁面底部設有「編輯」按鈕,點擊後開啟編輯模式,按鈕下方則是目前已套用的節能設定摘要,方便快速確認各項選擇,如圖十四所示。



圖十三：節能設定。圖十四：省電效能展示。

4.1.10. 通知設定

通知設定介面讓使用者能依需求設定各類通知項目,包括用電異常通知(電流過載或設備故障)、環境警告提醒(溫度或濕度過高)以及設備狀態警告(感測器異常或離線)。使用者可針對每項通知選擇是否啟用,並可設定偏好的通知方式。介面右上角的調整圖標提供異常閾值的自訂功能,

讓使用者能自行輸入觸發提醒的判定條件。頁面底部則設有「通知歷史記錄」按鈕，可快速進入歷史紀錄頁面查看先前的通知內容。

本系統提供通知歷史記錄介面，會自動保存各類型的正常與異常通知，讓使用者能方便地回顧與追蹤過往事件。介面右上角設有篩選按鍵，使用者可依指定的日期範圍篩選通知內容，以快速查找所需資訊。頁面底部則呈現目前的頁數與通知總量，並提供分頁切換功能，便於瀏覽大量記錄。

5. 結論

本研究成功整合 IoT 裝置、環境感測技術與機器學習模型，建構一套能自動分析室內環境並調節家電設備的智慧節能控制系統。透過 LifeSmart 感測器與智慧插座取得的環境與用電數據，搭配 CatBoost 迴歸模型預測冷氣與風扇之最佳設定值，使系統能在兼顧使用者舒適度的情況下實現節能化運作。以 PMV 為熱舒適核心推論參數，使調控決策更貼近真實人體感受，提升模型應用的實務價值。

系統於 Web 與 Android 前端均具備完整操作介面，包含裝置控制、用電監測、節能設定、異常警示與帳戶管理等功能，使使用者可即時掌握室內狀態並遠端管理設備。結果顯示，本系統具備良好的穩定性、擴充性與實際應用潛力，尤其適合部署於智慧校園與智慧建築中。未來可進一步結合增強式學習、自動排程策略或更多環境變數，以提升模型精準度並擴展系統應用範圍。

6. 專題作品影音平台網址

<https://youtu.be/M2V6qQufzOQ>

參考文獻

- [1] S. Shalev-Shwartz and S. Ben-David, "Understanding Machine Learning: From Theory to Algorithms," Cambridge University Press, 2014.
- [2] P. Domingos, "A Few Useful Things to Know About Machine Learning," Communications of the ACM, Vol. 55, No. 10, pp. 78–87, October 2012.
- [3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," Future Generation Computer Systems, Vol. 29, Issue 7, pp. 1645–1660, September 2013.
- [4] P. O. Fanger, "Thermal Comfort: Analysis and Applications in Environmental Engineering," Danish Technical Press, Copenhagen, 1972.
- [5] K. E. Charles, "Fanger's Thermal Comfort and Draught Models," IRC Research Report RR-162, National Research Council of Canada, October 2003.
- [6] S. Lin, S. Wei, C. Huang, and W. J. Chen, "Thermal Comfort Study of an Air-Conditioned

Presentation Room in Taiwan," Journal of Architecture, Vol. 65, pp. 125–138, 2008.