**(二) 研究計畫之背景與目的**

1. **背景**

現今雲端運算技術發展成熟，已經成為許多企業佈署應用服務的主要方法。隨著晶片製造科技微型化，單一使用者可以同時擁有多個裝置聯網存取服務。也因為網路技術的普及，用戶可以來自世界各地與伺服器連結使用，大程度的帶來便利性。然而，大量的用戶導致伺服器發生意外的故障與停機，影響用戶的使用體驗，尤其是高風險應用更為顯著。因此，正需要導入智慧化的故障預測系統防止無預期的事件發生，減少意外帶來的高額成本及損耗。

伺服器的狀態監控可以藉由溫度、硬體資源使用率、系統日誌等紀錄來觀測。雖然現今已有許多基於深度學習的方法來辨認異常狀態，但仍然無法有效的解決此問題。一般的深度類神經網路模型會受限於樣本不均衡的問題，使得模型傾向於預測樣本比例大的類別來得到最好的損失。半監督學習方法使用自動編碼器學習對於正常狀態建模，在實際使用時判定與正常分布的差距來辨別異常。而正常模式卻會頻繁的變換導致模型失準，需要時常離線重新學習來維持準確度。為了能讓雲端應用服務維持正常營運，減少因故障帶來不好的經濟損失，我們為台灣恩智浦半導體股份有限公司(NXP)量身訂做一套雲端應用服務異常檢測系統。半導體工廠在對晶片加工與測試時，對於一片複雜設計的晶片會產生大約數萬筆參數。因此，資料的上傳與存取都必須要確保伺服器隨時處於正常狀態，並在問題發生時能儘快啟動相關機制防止用戶無法及時使用資料。在大公司的分佈式系統中，為了確保每項服務彈性佈署的特性，管理者會建立多項虛擬機維持營運，並視情況執行擴充或遷移。然而，一個異常檢測系統需要監測如此多樣且複雜的系統是具有挑戰性的。我們必需要考慮到異常檢測模型的適應性，設計一套對於隨時間變動的規則，能夠以最少的資源重新訓練與適應，大程度的減少維護的複雜流程的全方位系統健康檢測系統。系統可以透過內部的檢測機制判定模型可否維持在線服務，在必要時也能輔以人工判讀。本計畫提出「基於數據驅動元強化學習的應用服務之計算資源異常偵測與預測」乙案，分三年期程，逐年分別提出 (1) 基於強化學習的伺服器異常檢測、(2) 元強化學習的自適應流程和(3) 基於元策略流程的主動異常檢測系統等三項的基於數據驅動元強化學習的應用服務之計算資源異常偵測與預測之研究解決方案，以三年期程解決台灣恩智浦公司在生產過程所遭遇的棘手問題。

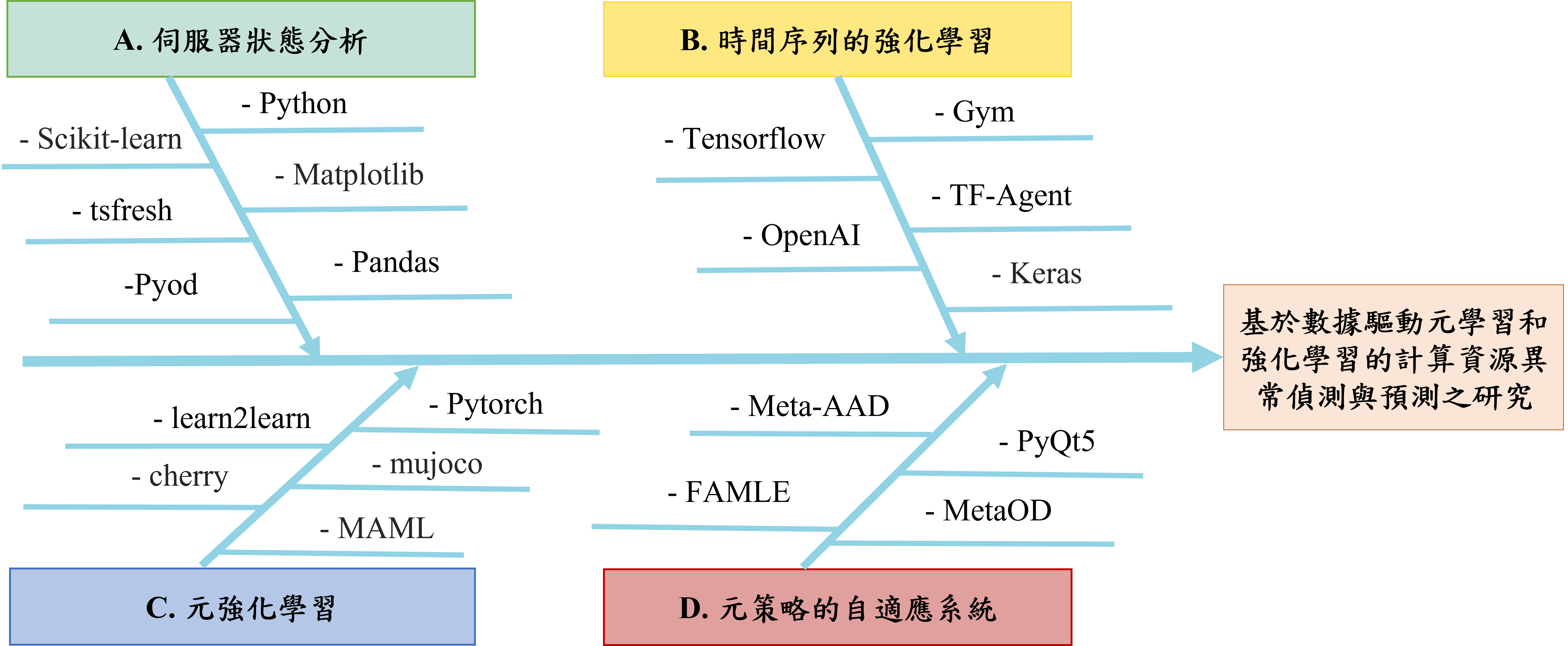


Fig. . 基於數據驅動元強化學習的應用服務之計算資源異常偵測與預測之研究

如Fig. 1所示，基於數據驅動元強化學習的應用服務之計算資源異常偵測與預測之研究所涵蓋技術包含了(A)伺服器狀態分析、(B)時間序列的強化學習、(C)元強化學習和(D)元策略的自適應系統。本計畫提出「基於數據驅動元強化學習的應用服務之計算資源異常偵測與預測之研究」乙案，將會針對(C)和(D)的部份聚焦於主動式的元策略異常檢測流程的優化。如此一來本計畫便能解決因為雲端環境變化而發生模型失準以及減少重新佈署模型之成本。

本案預計與台灣恩智浦公司合作進行三年期改善的解決方案。首先該公司在僅使用Zabbix Server監控公司雲端服務的健康狀態，並且使用靜態閥值作為系統發生警訊的依據。此方法因為對於某些時常處於高度使用的應用服務會時常收到警示郵件，但經檢查後卻是處於正常狀態。然而，當故障發生時卻是無預警且沒有任何徵兆。因此，為了讓公司能夠對於系統的故障能夠及早的準備，計畫第一年以深度強化學習演算法進行雲端應用服務的異常檢測，大幅減少管理者因為假警報或是無預警的故障帶來的損失；其次台灣恩智浦公司的多個應用服務性質不同，需要為每個應用服務制定一個預測模型。然而，多個模型訓練的成本將難以估計。所以第二年計畫我們將強化學習中加入MAML的訓練流程，減少不同應用服務之間更新模型的複雜性；最後為了維持檢測系統的持續性，我們期望能以最少的人工干預來維護在線的檢測模型。第三年計畫以Meta-AAD的檢測流程建立一個全自動化的系統，再加入FAMLE執行快速適應主動檢測無預期的故障，提升台灣恩智浦公司雲端服務的穩定性。

1. **目的**

本計畫的目的是實現在數位化的時代下以資料驅動結合人工智慧的力量，建立資料中心集群服務異常檢測系統，減少非預期的故障帶來的經濟損失。因此提出「基於數據驅動元強化學習的應用服務之計算資源異常偵測與預測之研究」乙案並分三年期程執行，逐年分別提出(1) 基於強化學習的伺服器異常檢測、(2) 元強化學習的自適應流程和(3) 基於元策略流程的主動異常檢測系統等三項基於數據驅動元強化學習的應用服務之計算資源異常偵測與預測之研究的解決方案。如Fig. 2所示，第一年期程預計導入基於強化學習的伺服器異常檢測，做為台灣恩智浦公司伺服器故障分析與檢測的解決方案，運用強化學習演算法辨識複雜的雲端故障現象。如Fig. 3所示，第二年期程預計導入元強化學習的自適應流程，可為台灣恩智浦公司維護在線異常檢測系統，並導入Meta Learning自適應方法加速模型更新流程。如Fig. 4所示，第三年期程預計導入基於元策略流程的主動異常檢測系統，為台灣恩智浦公司伺服器管理維護部門開發一個主動學習的異常檢測系統，在訓練部分使用Meta-AAD進行主動學習，並加入不同系統狀態下的向量，並在在線預測時選擇最適合模型執行任務。



Fig. . 基於強化學習的伺服器異常檢測之流程

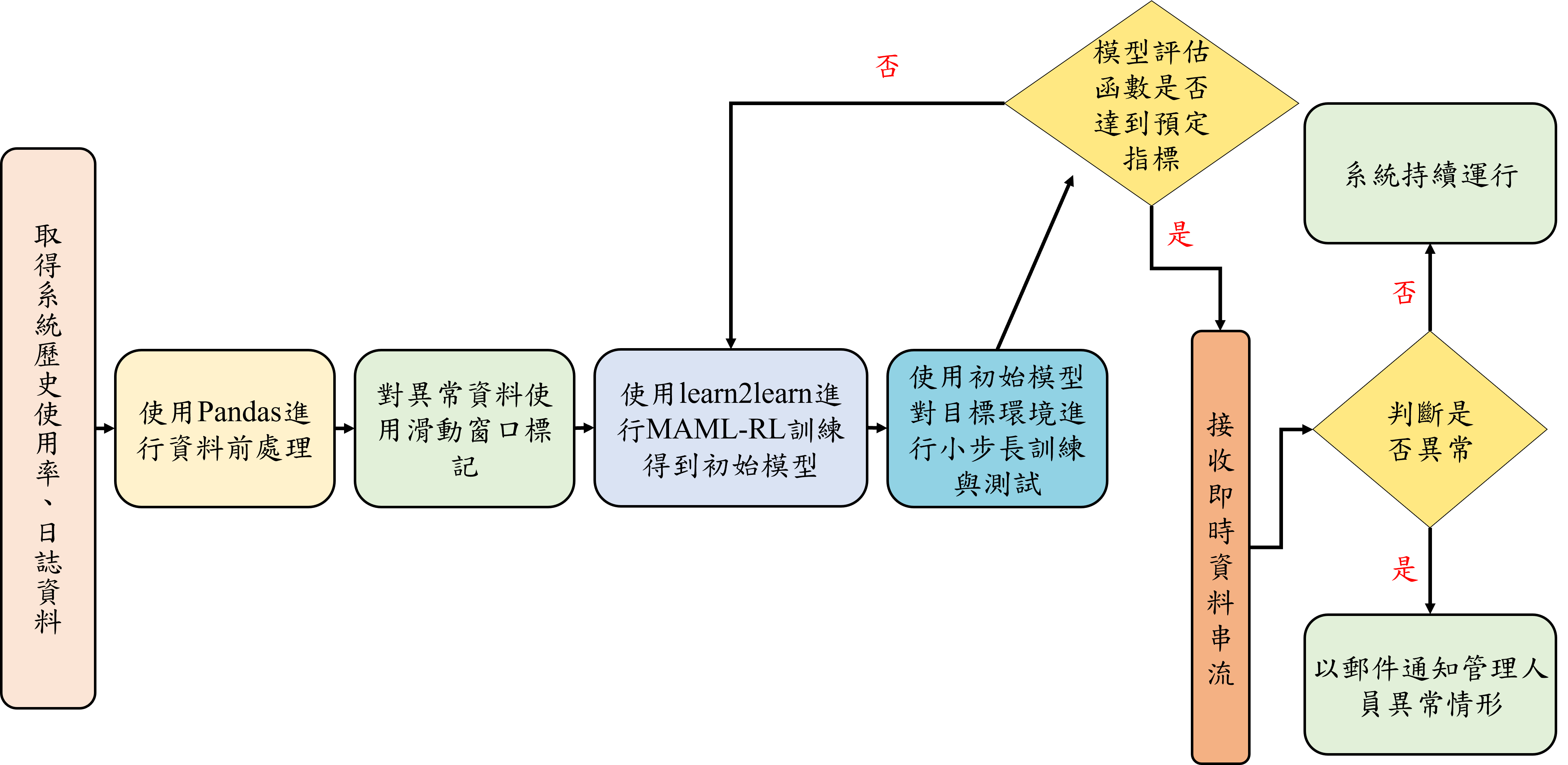


Fig. . 元強化學習的自適應之流程

1. **重要性**

如Fig. 5所示，本計畫的重要性是實現在台灣恩智浦公司的雲端服務維護上的進步。使用目前適應性較佳的MAML演算法，考慮到雲端應用的複雜性，還加入了主動學習方法Meta-AAD與FAMLE演算法。如此一來，在適當人工的介入下，主動學習更能幫助模型學習到更正確的資訊，也能充分理解模型認知的異常內容與微調方向。再來，也可以模擬不同故障所產生的使用率資料來幫助在線預測系統選擇最適合的模型進行預測。

Fig. 5. 基於數據驅動元強化學習的應用服務之計算資源異常偵測與預測之研究之整體架構說明

近年來因為工業4.0的推動，工廠逐漸以智慧製造為目標，使生產流程能更加的智慧化。然而，智慧化的設備雖然大幅提升生產的效率，不過若產品製造的過程需要高精密度。機台的動作精準度將會格外重要，只要有些微的誤差則容易造成產品的損失。讓生產流程更加智慧化的做法主要是將需要使用到人力的部分能夠盡量以機器做為取代，一方面能帶來有效率的工作，另一方面能夠提升品質檢測時的穩定性。然而，智慧化的品檢設備確實帶來生產流程的效率，不過產品製造的過程仍需要高精密度。隨著時代更迭，人工智慧的市場需求水漲船高。也因為市場的需求，許多科學家及工程師紛紛投入人工智慧的行列，於是各式新技術也應運而生。YOLOv1演算法[12]在發表的當時即造成轟動，因為該演算法是以單階段物件偵測的模式進行影像辨識及物件分類，因此不但可以即時地偵測物件，在當時的準確性表現也相當傑出。於隔年由YOLOv1演算法改良的YOLOv2演算法[13]，其改良目的在於提升即時性及準確性，因此透過批次標準化及錨框機制達到兩者的提升。時至今日，因YOLOv2演算法的模型以較為輕巧的架構組成，於是YOLOv2演算法在即時性的效果仍是YOLO演算法系列中不可動搖的地位。於是我們初步嘗試使用YOLOv2演算法應用於本計畫的研究內容。透過各種調整參數的方式，反覆測試YOLOv2演算法的模型是否具有良好的準確性。在使用晶片製程的影片測試時，造成漏檢的情形相當嚴重。經評估後僅有約五成的準確度，將無法達到台灣恩智浦公司期望的準確度。因此我們將焦點轉移至以YOLOv2演算法為基礎進行改良而誕生的YOLOv3演算法[14]。

YOLOv2演算法與YOLOv3演算法之間最大的區別在於YOLOv3演算法加入殘差網路[15]的概念。殘差網路可說是深度學習的最大助力，在沒有殘差網路的理論出現時，深度學習在建立模型時容易因為梯度消失的問題導致模型深度無法加深。在殘差網路出現後，紛紛有人透過將殘差網路的概念引進深度學習中。來自世界各地的學者經過實驗後發現殘差網路當中殘差連接的概念對於深度學習是有相當大的影響力，透過殘差連接的方式使梯度能夠不斷向下傳遞而不容易消失。因此現今許多有名的深度學習模型都使用到殘差網路的概念。透過相同影片進行測試後發現，YOLOv3演算法的模型進行評估後可得到約八成的準確度，與YOLOv2演算法的模型相比之下提升約三成的準確度。不過以實際進行產線的品質檢測而言，八成的準確度仍是差強人意的。我們同樣測試YOLOv4演算法[16]的效果，僅針對台灣恩智浦公司所提供的資料而言，會因為影像品質與內容的背景複雜程度等因素造成一定程度的影響。雖然對於檢測物件的種類信心程度極高，不過將正常晶片判斷成損壞晶片的頻率偏高，在誤報率的考量下將不利於產線的品質檢測。

最後我們決定使用於2020年問世的YOLOv5演算法。YOLOv5演算法與YOLOv4演算法之間的差異在於YOLOv5演算法於網路結構一開始使用Focus結構。目的與特徵壓縮相同，只是該做法是以拼接的方式將特徵圖拆分為四等分，並將四個等分串接在一起，實際上特徵的訊息並未受到影響，而以壓縮的方式則會失去一些影像的訊息。另外自動錨框計算則是針對不同資料集進行訓練時可為資料集設定初始錨框大小，使錨框大小適合於使用的資料集。最後是自動影像縮放的技巧，進行影像推論的過程，主要會將影像設定為長寬等長的大小。如果影像來源長寬比例較大時，則根據縮放係數將短邊形成的填充影像大幅減少，減少推論所需的計算量，增加推論的效率。以相同影片進行測試時，發現YOLOv5演算法的模型能夠達到零誤報的表現，在晶片製程的測試影片能夠達到100% 的準確度。不過由於在Workstation上執行物件偵測可能會造成接收即時影像時的影像延遲。於是我們提出採用較新版本的YOLOv5演算法加上嵌入式平台NVIDIA Jetson Nano與嵌入式平台NVIDIA Jetson Xavier NX的部署[17]來嘗試解決台灣恩智浦公司對於目前問題所遇到的瓶頸。

1. **國內外相關研究**
2. 金大開發高粱酒酒標檢測防偽技術 辨識正確率高達95%，2021-8-6。
3. **重要參考文獻**
4. C. Wang, H. Mark Liao, Y. Wu, P. Chen, J. Hsieh and I. Yeh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 1571-1580.

**(三) 研究方法、進行步驟及執行進度**

1. **研究方法與原因**

**研究方法**

**第一年期程**

**（1）異常資料標記**

首先我們使用台灣恩智浦半導體股份公司用於半導體封測生產線資料的上傳、搜尋與運算的應用服務Oplus，資料集為每三分鐘接收一次資料。其中，包含了CPU、Memory、Disk Queue等資料以及不同編號的虛擬機(代號1V、2V、3V、4V......等)。我們主要觀察系統的CPU與Memory使用率資料作為主要特徵，如Fig. 4所示。

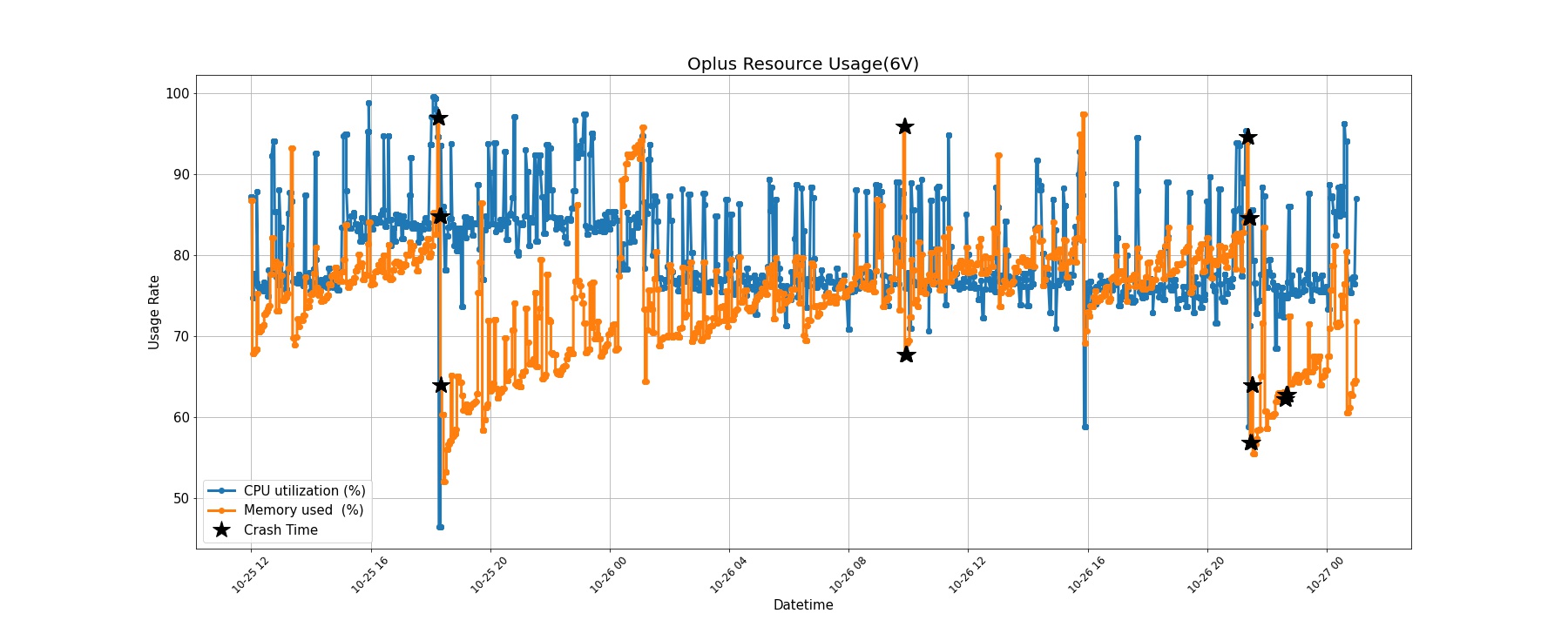


Fig. . Oplus資源使用率與異常時間標記

對於真實的異常時間是以一固定寬度的窗口標記。在實際的二元數字標記上我們希望當檢測模型遇到異常的前兆時發出警告。因此，資料集在異常區間內都標記為1，其餘正常資料我們標記為0，如此一來窗口內只要出現一個異常點的話即可認定此窗口可能出現異常。標記狀況如Fig. 5所示。

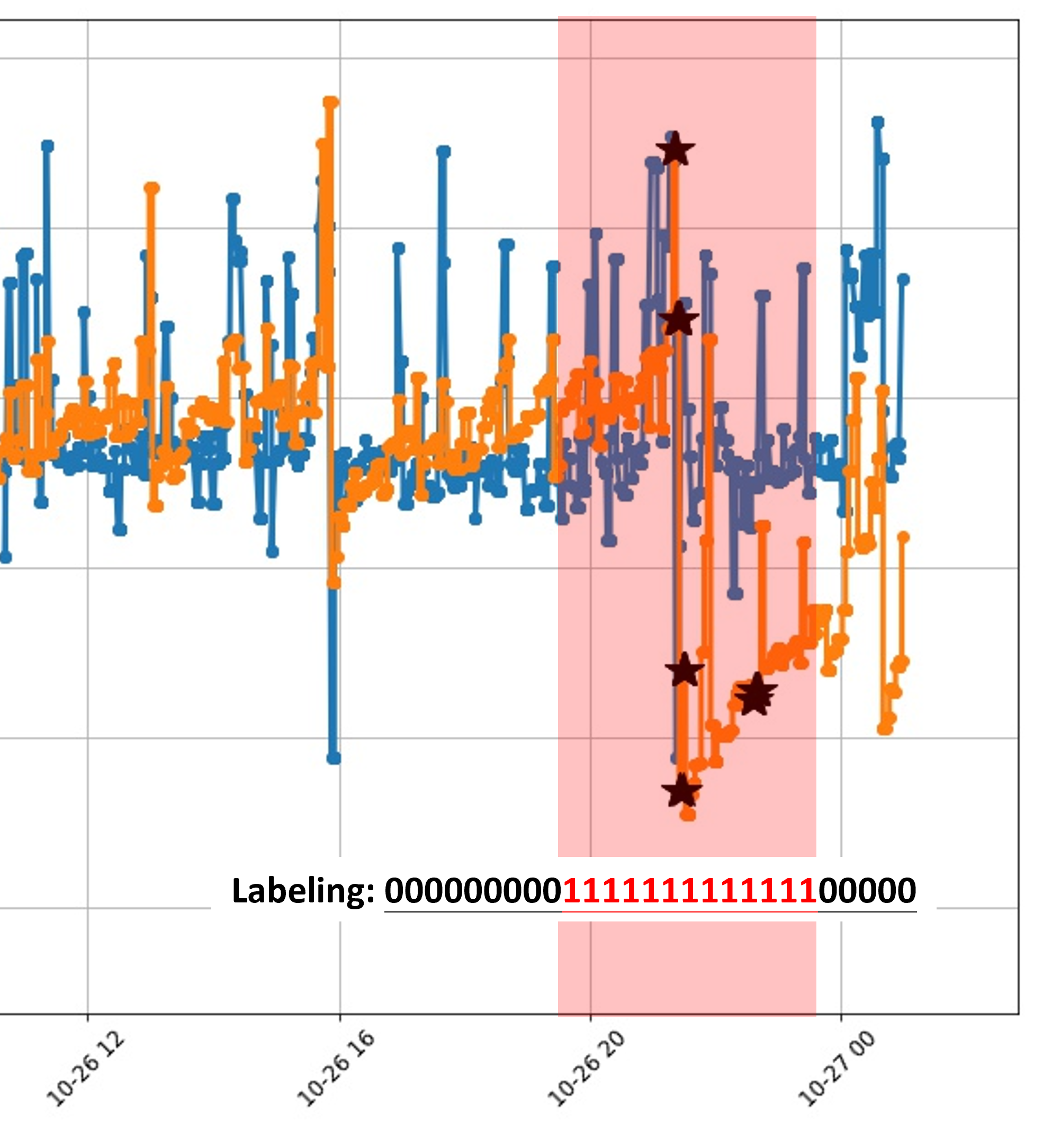


Fig. . 異常窗口的標記方法

**（2）建立深度強化學習環境**

本計畫使用RLAD[]方法建立強化學習環境，首先，我們將時間序列使用固定的滑動窗口擷取資料作為環境狀態(State)，並將動作空間(Action Space)限制為兩個離散狀態:0(正常)、1(異常)。將環境狀態作為輸入資料輸入策略網路(Policy Network)輸出代理人的決策，並且有一個概率決定代理是否遵從策略網路提出的建議。輸出的決策會與資料集的標籤計算回饋(Reward)，在此我們設定回饋函式為。最後，會有一個記憶模組(Replay Memory)儲存此轉移狀態，如Fig. 6所示。

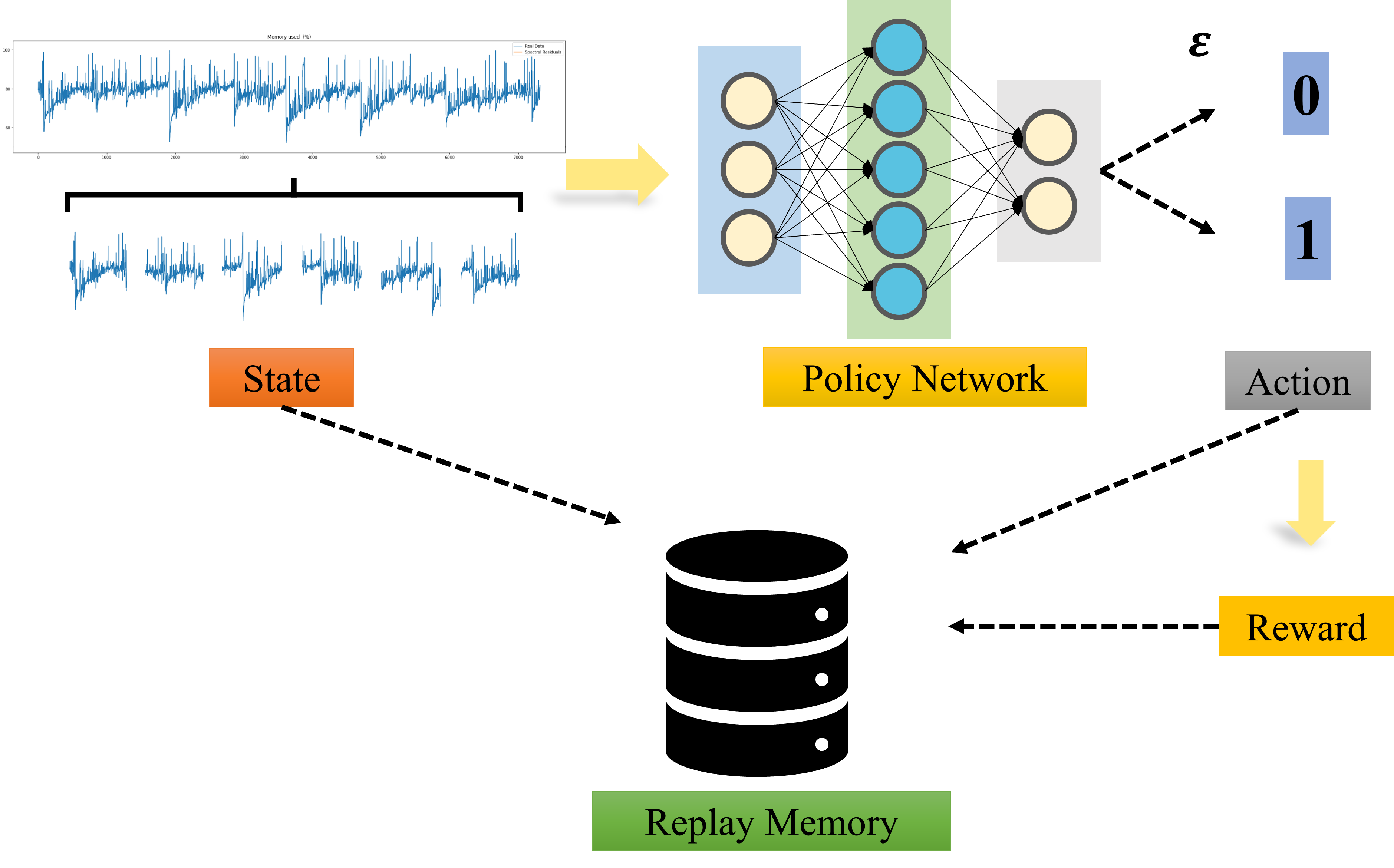


Fig. . 強化學習流程圖

我們額外加入一個隨機抽取記憶中的狀態與本身的輸出值執行梯度下降法更新參數，並在間隔固定的迭代次數後複製模型參數作為後進行下一次的回合訓練，如Fig. 7所示。

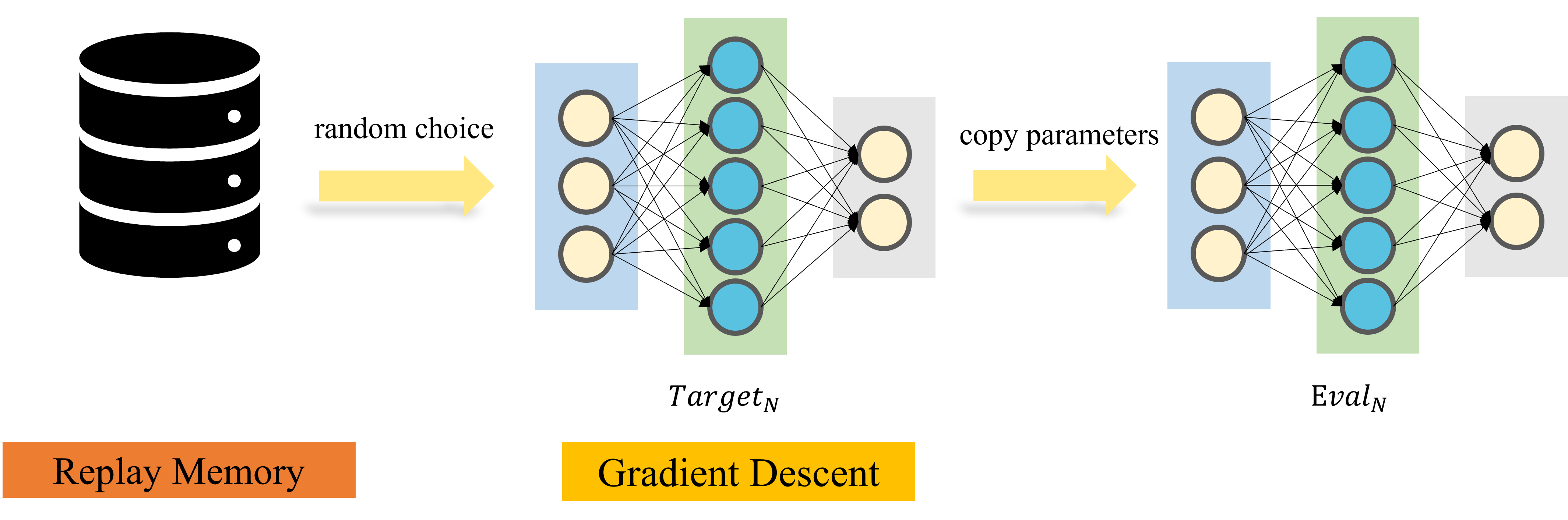


Fig. . 策略網路更新流程

**（3）策略模型評估**

本計畫以PyTorch做為訓練框架，過程會觀察每一個迭代累加的Reward數值，並透過觀測趨勢修改環境設定的獎勵與懲罰機制，如Fig. 8所示。



Fig. . 強化學習訓練過程之Reward趨勢

我們使用Precision、Recall、F1-score三項指標評估模型是否達到可以佈署的程度，公式如下。

|  |  |
| --- | --- |
|  | (1) |

|  |  |
| --- | --- |
|  | (2) |

|  |  |
| --- | --- |
|  | (3) |

**（4）佈署模型執行在線預測**

經過評估可使用之模型將會佈署至台灣恩智浦半導體股份公司的運算集群。該公司目前是使用Hadoop與Spark建立集群系統，使用Zabbix Server監控各個虛擬機的資源使用率。系統設定固定時間會將資料匯入HDFS(Hadoop Distributed File System)。本計畫使用Spark streaming將資料轉換成DStream(discretized stream)形式輸入在線預測模型進行故障預測。(補圖)

**第二年期程**

**（1）時間序列特徵擷取**

本期程針對時間序列做進一步的特徵擷取。首先，原始的時間序列包含了複雜的特徵空間，導致強化學習的訓練效果不佳。因此，我們對原始資料的統計域與頻率域進行分析，尋找對於模型的最佳特徵映射。

我們使用TSFEL(Time Series Feature Extraction Library)[]進行分析。首先，我們透過Python Pandas讀取不同編號的虛擬機(代號1V、2V、3V、4V......等)的CPU、Memory使用率資料。再來，使用TSFEL套件內擷取出統計域特徵，如最大值與最小值，梯度等並觀察各項數值與折線圖，如所示。(實驗作圖)。最後，將擷取特徵與異常標記以CSV檔案儲存。

**（2）建立元學習子任務**

本計畫融合了強化學習與Meta Learning演算法，提升強化學習對於不同的環境下的適應性。Meta Learning將資料集拆分為不同的小型分類任務並分別執行訓練與分類，並得到一個初始化模型去適應當前的任務。對於雲端系統的不同的應用服務下需要不同模型分別專注於分類當下環境的異常，若使用一般機器學習方法勢必會造成過多的時間成本。因此，我們運用該演算法的優勢減少管理人員新增或修改某項服務的檢測系統。

對於Meta Learning的強化學習，我們將Environment視作一個小型的分類任務，並小幅度修改強化學習環境作為元學習的子任務。原本作為靜態的強化學習環境修改為僅有一次的互動，在透過多執行緒分別建立多個環境訓練。Action為強化學習中的Agent判定是否異常，並將Reward設定為(TP, TN, -FP, -FN)。我們採用DNN建立作為Policy Network，Policy Network會接收環境狀態並輸出機率分布，再由分布中採樣一個Action。每次Task運作結束後會將State、Action、Reward資料儲存至Replay Memory儲存。(多執行緒->環境)

**（3）建立MAML演算法**

MAML(Model-Agnostic Meta-Learning)是一種與模型本身無關的學習演算法，目標在於多種任務中學習並從中得到任務的本質。其中，包含了Inner Loop和Outer Loop。Inner Loop負責產生多個Task，每個Task會採樣一小部份的時間序列訓練並保存經過梯度下降的參數。再來，Outer Loop將訓練後的參數進行測試，記錄損失並使用適合的優化函式更新最終網路參數。

本研究使用learn2learn套件與python語言開發。首先，我們的Inner Loop建立多個執行緒用於同時在多個強化學習環境下執行，並為每個環境建立一個Task與策略網路()。再來，將每個環境產生的State、Action、Reward、Loss及更新後的策略網路參數()儲存至Iteration Replay。在Outer Loop部分，我們選擇TRPO找出最佳策略。TRPO演算法重複使用策略的Trajectory增加樣本的使用率，也能確保強化學習在訓練的過程不會因為策略的改變影響模型的學習效果，該演算法也透過劃定一個可信任的策略學習區域，保證策略學習的穩定性和有效性。流程圖如Fig. 9所示。

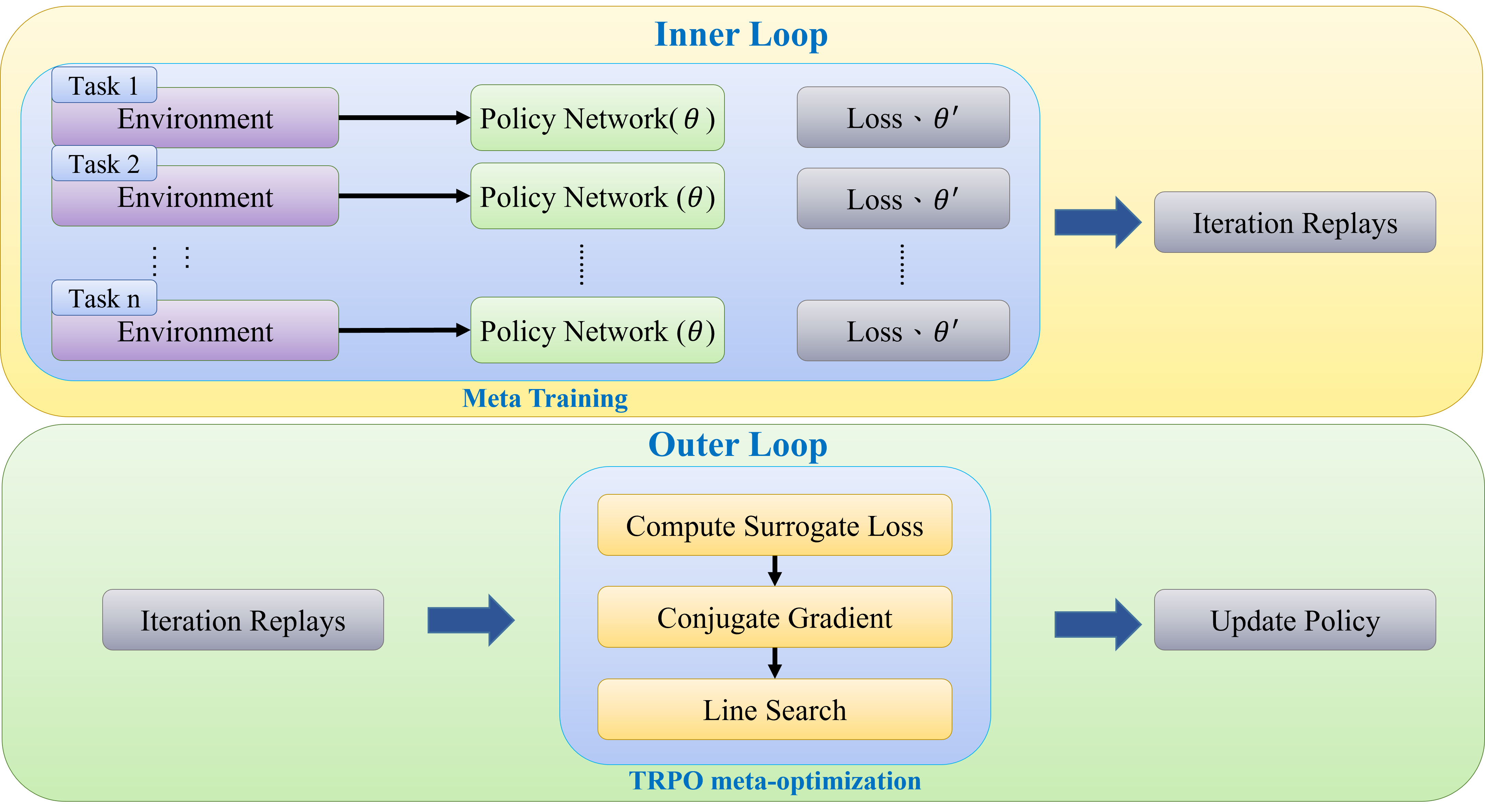


Fig. . MAML-RL訓練流程圖

**（4）訓練元強化模型**

本步驟首先使用OpenAI gym套件註冊資源使用率環境作為本實驗的Environment，並設置Inner Loop之超參數如下表。

Table . MAML超參數設置

|  |  |
| --- | --- |
| Hyperparameter | Value |
| Hidden layer | [128,128,128] |
| Adapt learning rate | 0.5 |
| Number of iterations | 100 |
| Meta batch size | 10 |
| Number of workers | 10 |
| cuda | 1 |

每個Iteration設置記錄檔紀錄Reward與目前最終模型的準確度，並設置Tensorboard觀察訓練過程趨勢，如Fig. 10所示。



Fig. . Tensorboard觀察訓練過程

**（5）模型自適應**

上步驟的初始化模型

**第三年期程**

**（1）可轉移的元特徵擷取**

在元學習的過程我們橫跨了不同屬性的資料集，雖然在資料前處理部分已有基於統計域的特徵，但此方法過度依賴原資料，當資料轉變時特徵的取得會變得毫無意義。為了能夠在不同資料提取一個可轉移的元特徵，更精確的評估一個異常資料的程度，我們集成了數個無監督學習模型作為前處理的模組。

本步驟使用了PyOD 套件集成了OCSVM(OneClass SVM)、iForest(Isolation Forest)以及LODA(Lightweight on-line detector of anomalies)[]無監督異常檢測模型，使用模型輸出的異常分數作為資料集的狀態映射。另外，我們還加入了樣本與異常資料之間的距離作為特徵。經過元特徵擷取後儲存成CSV檔案作為元學習訓練使用。

**（2）異常情況模擬**

本步驟

**（3）基於主動學習的異常檢測**

主動學習、排名異常實例、領域專家確認

**（4）FAMLE情境模擬訓練**

本FAMLE<https://www.eginnovations.com/blog/server-performance-monitoring/>參考遷入以下的故障模擬情境

**（4）嵌入式平台進行推論**

本期程

**進行步驟**

**第一年期程**

(1) 從原始使用率以虛擬機編號、硬體名稱、時間提取重要資料。(2) 對時間序列資料進行滑動窗口前處理。(3) 對異常資料進行固定長度窗口標記。(4) 切分資料為訓練集與測試集。(5) Pytorch訓練環境建置。(6) 強化學習GYM環境設置。(7) 強化學習模型訓練。(8) 使用測試集測試策略網路與評估性能。(9)將策略網路佈署至Hadoop&Spark集群。(10) 從HDFS接收Zabbix Server即時串流資料預測。(11) 模型預測即將發生異常時通知管理者。

**第二年期程**

(1) 以原始資料使用TSFEL提取統計域特徵。(2) 將原資料映射為統計特徵並加入原資料標記。(3) learn2learn環境安裝與設置。(4) 修改強化學習GYM環境為元學習產生子任務。(5) 使用多執行緒建立元學習子任務。(6) 使用TRPO優化更新網路參數。(7) 提取初始模型與監測目標的資料進行小步長訓練。(8) 評估最終模型性能(9) 將最終模型佈署至監測目標。

**第三年期程**

(1) 使用無監督異常檢測器輸出異常分數。(2) 計算樣本與異常樣本的距離。(3) 將資料集映射為檢測器分數與異常距離。(4) 模擬異常狀況生成資料。(5) 針對合作廠商進行技術轉移。

**執行進度**

**第一年期程**

(1) 訓練資料與驗證資料異常標記與資料設置。(2) 訓練強化學習模型的參數設置。(3) 訓練過程之設定參數有效性測試。(4) 最終策略模型有效性測試。(5) 模型佈署於集群系統有效性測試。(6) 在線預測資料速度與運算效率測試。(7) 連結串流資料的穩定性測試。(8) 預測模型準確度測試。

**第二年期程**

(1) 時間序列特徵擷取之有效性與方法選擇測試。(2) 元強化學習GYM環境之有效性測試。(3) 訓練元學習環境測試。(4) 多執行緒建立子任務的有效性測試。(5) 訓練元學習超參數設定有效性測試。(6) 優化函數有效性測試。(7) 模型自適應效益與時間測試。

**第三年期程**

(1) 異常檢測器異常分數有效性測試。(2) 元特徵有效性測試。(3) 異常狀態模擬與資料蒐集。(4)(11) 針對合作廠商進行技術轉移。

**二、預計可能遭遇之困難及解決途徑**

**第一年期程**

1. 原始時間序列特徵空間過於複雜，可能對強化學習效果有影響。
2. 強化學習的優化方法過於單調，可能需要測試不同優化策略達到回饋最大化
3. 強化學習會是一個長時間的模擬過程，需要在有限的時間與硬體資源下權衡出最佳模型。
4. 佈署至集群環境時要轉換資料型態以符合PySpark的運算模式，在撰寫在線預測程式時要考慮到轉換的方便性。

**第二年期程**

1. 特徵擷取時有許多方法可以參考，但要找出橫跨所有資料集都可以顯著的方法。因此在特徵擷取方法測試期間應要考慮不同資料間的共通性。
2. 元學習尚處於發展階段，因此線上資源較為缺乏需要自行修改現有函式庫以新增指定功能。
3. 元學習有強大的適應性是一項特點，但要與一些基準比較該模型在此領域的準確度做比較與評估，再執行下一步佈署流程。

**第三年期程**

1. 本期程提供的三個異常檢測器為目前較先進的方法，再找尋其他非線性方法可以改善異常分數的可靠性。
2. 異常狀態模擬資料可以體現常見故障的情境，但也需要透過領域專家分析並找出一些隱藏故障才可以有效的模擬異常。因此，與管理者的溝通與設計方法是很重要的部分。

**(四) 預期完成之工作項目及成果**

**ㄧ、本研究計畫預期完成之項目**

本計畫三年期程的甘特圖，如Table 4所示。

Table 4. 三年期程甘特圖Gantt Chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 月數  工作 | 三年計畫 | | | | | | | | | | | |
| 1-3 | 4-6 | 7-9 | 10-12 | 13-15 | 16-18 | 19-21 | 22-24 | 25-27 | 28-30 | 31-33 | 34-36 |
| 月. | 月. | 月. | 月 | 月. | 月 | 月. | 月 | 月 | 月 | 月. | 月 |
| 基於強化學習的伺服器異常檢測 |  |  |  |  |  |  |  |  |  |  |  |  |
| 第一期各項效能測試與人員訓練 |  |  |  |  |  |  |  |  |  |  |  |  |
| 元強化學習的自適應流程 |  |  |  |  |  |  |  |  |  |  |  |  |
| 第二期各項效能測試與人員訓練 |  |  |  |  |  |  |  |  |  |  |  |  |
| 基於元策略流程的主動異常檢測系統 |  |  |  |  |  |  |  |  |  |  |  |  |
| 第三期各項效能測試與人員訓練 |  |  |  |  |  |  |  |  |  |  |  |  |
| 針對合作廠商進行技術轉移 |  |  |  |  |  |  |  |  |  |  |  |  |
| 撰寫研究報告和公佈研究結果 |  |  |  |  |  |  |  |  |  |  |  |  |
| 預期的進度累進比例 | **15** | **35** | **40** | **50** | **60** | **70** | **75** | **80** | **85** | **90** | **95** | **100** |

**第一年期程**

(1) 完成資料清洗與提取重要資料。(2) 完成窗口資料處理與異常資料標記。(3) 完成Pytorch訓練環境建置。(4) 完成強化學習GYM環境設置。(5) 完成強化學習模型訓練與參數設置的有效性測試。(6) 完成模型評估。(7) 完成檢測模型佈署以及有效性測試。(8) 完成資料串流連接測試。(9) 完成集群模型佈署。

**第二年期程**

(1) 完成時序資料特徵擷取。(2) 完成learn2learn元學習訓練環境設置。(3) 完成元強化學習GYM環境設置。(4) 完成元強化模型訓練。(5) 完成元強化模型有效性測試。(6) 完成元強化模型適應性測試。(7) 完成佈署元強化模型

**第三年期程**

(1) (11) 完成針對合作廠商進行技術轉移。

**二、對於學術研究、國家發展及其他應用方面預期之貢獻**

**第一年期程**

(1) 提出雲端複雜資料的檢測技術。(2) 提出時間序列異常檢測的解決方案。(3) 技術通用性高可使用於不同集群。(4) 特殊的資料處理方式有助於其他專案之參考做法。(5) 模型有效結合集群增進效率

**第二年期程**

(1) 解決資料特徵過於複雜的技術。(2) 提出元強化學習技術應用在伺服器異常檢測。(3)解決一般模型適應性的問題。

**第三年期程**

(1)

**三、參與之工作人員，預期可獲之訓練**

**第一年期程**

(1) 學習資料處理技術。(2) 學習強化學習演算法與程式撰寫。(3) 學習Pytorch使用方法。(4) 實作資料科學應用。(5) 理解硬體加速的重要性。(6) 熟知Hadoop與Spark基礎架構。(7) 學習深度學習的概念與技術。

**第二年期程**

(1) 熟知開源軟體開發與使用技術。(2) 獲得軟體工程相關概念與技術。(3) 學習特徵擷取技術與統計相關知識。(4) 學習元學習的概念與技術。(5) 熟知強化學習優化策略之演算法。(6) 學習資料增強的概念與技術。

**第三年期程**

(1) 熟知開源軟體開發與使用技術。(2) 獲得軟體工程相關概念與技術。(3) 智動化檢測的相關概念與使用技術。(4) 學習影像辨識的相關概念與技術。(5) 學習深度學習的概念與技術。