

**科技部資訊安全實務研發計畫  
『系統測試報告書』**

**System Integration and Testing Document  
Of  
CyberSecurity Practices and Research(CPR)  
Project**

**先進駕駛輔助系統之雲端輔助設計優化  
子計畫二：技術為基之智慧計算框架研發以支援先  
進駕駛輔助系統之研究與實作  
MOST 105-2221-E-006-141**

**計 畫 主 持 人：陳朝鈞 副教授  
(國立成功大學 製造資訊與系統研究所)**

**共 同 主 持 人：曾世邦 助理教授  
(大仁科技大學 資訊工程系)**

**國立成功大學 製造資訊與系統研究所**

**Institute of Manufacturing Information and Systems,  
National Cheng Kung University, Taiwan**

**2017/06/14**

# 目錄

<b>1. 測試目的與接受準則 (OBJECTIVES AND ACCEPTANCE CRITERIA)</b>	<b>3</b>
1.1 系統範圍 (SYSTEM SCOPE)	3
1.2 測試文件目的 (PURPOSE OF THIS DOCUMENT)	3
1.3 測試接受準則 (TEST ACCEPTANCE CRITERIA)	4
1.4 需求確認準則	4
<b>2. 測試環境 (TESTING ENVIRONMENT)</b>	<b>5</b>
2.1 操作環境 (OPERATING ENVIRONMENT)	5
2.2 硬體規格 (HARDWARE SPECIFICATION)	6
2.3 軟體規格 (SOFTWARE SPECIFICATION)	6
2.4 測試資料來源 (TEST DATA SOURCES)	7
<b>3. 測試行程、流程 (TESTING SCHEDULE AND PROCEDURE)</b>	<b>8</b>
3.1 測試時程 (TESTING SCHEDULE)	8
3.2 測試流程 (TESTING PROCEDURE)	8
3.2.1 SUBSYSTEMS VALIDATION	8
3.2.2 INTEGRATION TESTING	8
<b>4. 測試結果與分析 (TEST RESULT AND ANALYSIS)</b>	<b>8</b>
4.1 子系統一：巨量車載感測資料儲存與查詢系統	9
4.2 子系統二：巨量車載感測資料上傳系統	18
4.3 計畫測試結果	22

## 1. 測試目的與接受準則 (Objectives and Acceptance Criteria)

### 1.1 系統範圍 (System Scope)

子計畫二主要是進行「技術為基之智慧計算框架研發以支援先進駕駛輔助系統」的設計與實作。目標是利用收集來自許多車輛的車載感測資料，並透過網路上傳至雲端巨量資料儲存平台，接著藉由可處理巨量資料的智慧計算框架來針對資料量龐大，且具備高頻率特性的車載感測資料 (OBD2、GPS等) 進行儲存。為了達到上述系統之目標，我們建構出「結合Hadoop巨量資料處理技術的巨量車載感測資料上傳系統」，提供可對巨量車載感測資料進行高效率儲存的雲端儲存服務，並可支援各分項計畫所需之資料儲存服務。我們建構出「巨量車載感測資料上傳系統」，可將不同車輛上的車載感測資料進行上傳，將資料上傳至我們所建構之「巨量車載感測資料儲存與查詢系統」上，以提供各分項計畫後續的資料分析及應用。

子計畫二在總計畫內提供的功能包含：(1) 透過巨量車載感測資料上傳系統，將從許多車輛上所收集來包含有車載感測資料等資訊的檔案進行上傳，上傳到巨量車載感測資料儲存與分析平台上進行儲存。(2) 利用巨量資料儲存與分析平台將巨量的車載感測資料儲存至HDFS上，並利用Hadoop的巨量資料處理技術來對巨量的車載感測資料進行查詢。(3) 各分項計畫皆可透過通用型介面的資料存取應用程式介面 (RESTful API) 來支援各子計畫間資料溝通的服務，提供檔案儲存到HDFS、從HDFS取得指定條件之資料內容功能。

本文件將針對第一年度計畫所提出之需求規劃書，針對本系統第一年預計實現之目標進行下列各項子系統測試：

- (1) 子系統一：巨量車載感測資料儲存與查詢系統
- (2) 子系統二：巨量車載感測資料上傳系統

### 1.2 測試文件目的 (Purpose of this Document)

本文件主要建立「技術為基之智慧計算框架研發以支援先進駕駛輔助系統之研究與實作」的測試和整合計劃。在本系統進行整合之前，我們必須先確定子系統元件之單元測試是否已完成，並且著重於從需求文件和設計文件來進行整合系統測試(Integration Test)及接受度測試(Acceptance Test)。

本文件內容將依據系統需求規格書與系統設計文件，描述相關整合測試的相關計畫內容，並希望透過此文件之描述與實踐，達到順利進行測試工作之目的。

1.3 測試接受準則 (Test Acceptance Criteria)

本測試計畫需要滿足下列的測試接受準則：

- 本系統需要對所有列為優先(Critical、Important與Desirable)之需求進行完整測試。
- 測試程序需要依照本測試計劃所訂定的程序進行，所有測試結果需要能符合預期測試結果方能接受。
- 以測試案例為單位，當測試未通過時，需要進行該單元的測試，其接受的準則如第一項中所規定的相同。
- 解決系統中相互抵觸的元件與相關功能。

1.4 需求確認準則

本測試計畫依照以下各項需求的優先項目作完整的測試:

表 1 測試優先項目的需求確認表

編號	優先次序	說明
C	必要的 (Critical)	於系統完成時一定要具備，且必須經過確認才可以驗收。
I	重要的 (Important)	於系統完成時一定要具備，但允許部分功能上的妥協即可以驗收。
D	期望的 (Desirable)	於系統完成時不一定要具備，但如果完成可提高系統的價值。

## 2. 測試環境 (Testing Environment)

### 2.1 操作環境 (Operating Environment)

本計畫考量各分項計畫需要對大量的感測資料進行查詢，我們將透過巨量車載感測資料儲存與查詢系統來進行車輛感測資料 (OBD2及GPS等) 的儲存與雲端計算服務。因此，子計劃二「技術為基之智慧計算框架研發以支援先進駕駛輔助系統之研究與實作」的目標是能將巨量的車載感測資料儲存於雲端的資料庫中，並利用巨量資料處理技術來對HDFS中的巨量車載感測資料進行查詢。由於本計畫的測試情境為收集多部車輛的車載感測資料，而車載感測資料具有下列的特性，並相對應產生不同的問題：(1) 資料量大：不同車輛同時進行車載感測資料與車載診斷資料等的資料回傳，傳回各種資訊及影像等車載資料，導致資料庫會在短時間內接收到大量的資料。(2) 輸入與處理速度快：高頻率的資料收集造成資料量有很高的產生速率，為了避免資料處理速度不足造成查詢堆積導致系統崩潰，需要設計能處理巨量資料的資料管理框架，使得資料處理的速度能高於資料查詢的速度。(3) 資料多樣性：收集的資料中包含有車輛的位置、鏡頭影像、車速與引擎轉速等不同種類及型態的資料。

在本計畫的測試情境中，我們將聚焦於智慧車輛輔助系統的研發來達到駕駛輔助系統產品最佳化的目標。我們建置的雲端處理與儲存環境必須能處理大量使用者的車輛感測資料，而這些巨量資料必須要有可靠的雲端儲存系統來進行管理，並可透過統計技術或資料探勘技術來找出大資料中隱藏的知識，藉此提升資料價值做為駕駛輔助系統精進的關鍵應用。為了解決上述資料特性所產生的問題，我們必須建構一個能處理傳送大量資料到Hadoop巨量資料處理平台的系統。圖1為系統的架構圖，其詳細的子系統環境操作說明如下：

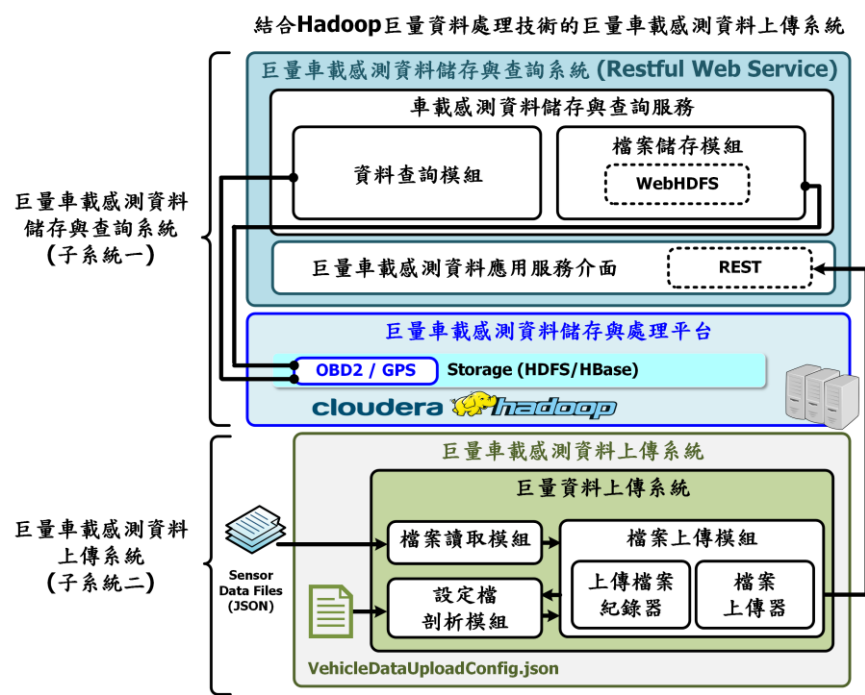


圖 1 子計劃二的子系統架構圖

子系統一為「巨量車載感測資料儲存與查詢系統」，主要的目標是要提供各分項計畫對巨量車載感測資料儲存與分析平台進行車載感測資料或所需資料的存取，並藉由 REST 通訊方式讓不同平台之間能進行溝通。

子系統二為「巨量車載感測資料上傳系統」，目的是將不同車輛上所收集到的車載感測資料上傳到目標雲端資料庫。由於目標資料庫的位置等需求會因為目標不同而有所改變，因此我們將常會改變的變數透過容易進行修改的 JSON 檔來做變更，透過解析這份檔案來得到欲進行上傳的目標位置以及上傳檔案位置等資訊。另外，此子系統中還包含了上傳與紀錄模組，紀錄上傳的時間與結果等資訊。

## 2.2 硬體規格 (Hardware Specification)

進行測試之巨量車載感測資料儲存與分析平台伺服器設備的硬體規格說明如下：

- Name Node（一台）：
  1. 處理器：Intel(R) Xeon(R) CPU E3-1220 v3 @ 3.10GHz
  2. 記憶體：32 GB
  3. 硬碟空間：2 TB
- Data Node（三台）：
  1. 處理器：Intel(R) Xeon(R) CPU E3-1220 v3 @ 3.10GHz
  2. 記憶體：32 GB
  3. 硬碟空間：  
Node2：10 TB  
Node3：13 TB  
Node4：6 TB
- 測試用主機：
  1. 處理器：Intel(R) Core(TM) i7-2600 @3.40GHz
  2. 記憶體：8 GB
  3. 硬碟空間：1 TB
  4. 電腦周邊：標準滑鼠、標準鍵盤

## 2.3 軟體規格 (Software Specification)

本系統開發與執行所需求的軟體說明如下：

- 規格：
  1. 作業系統：CentOS 6.7
  2. 開發工具：JDK 7、Tomcat 8、Axis 2、Eclipse、JavaScript、MySQL

## 2.4 測試資料來源 (TEST DATA SOURCES)

為了確認車載感測資料能有效且準確的上傳至我們所建構之巨量車載感測資料儲存系統進行儲存，我們透過實際的車輛道路駕駛來進行測試，我們收集的車載感測資料包含：

- (1) 油箱燃油狀況
- (2) 引擎附載量 (%)
- (3) 車輛引擎轉速 (rpm)
- (4) 車輛行駛時速 (km/h)
- (5) 引擎進氣溫度 (°C)
- (6) 引擎冷卻水溫 (°C)
- (7) 車輛位置經緯度 (度)

等共三十七項。測試資料的來源為測試人員實際於道路上進行車輛駕駛，並收集模擬不同的車況與情境時車載感測器所產生的實際車載感測資料。

### 3. 測試行程、流程 (Testing Schedule and Procedure)

#### 3.1 測試時程 (Testing Schedule)

- 時程
  - ✓ 子系統一「巨量車載感測資料儲存系統」程式開發 (2016/08/01 ~ 2016/12/20)
  - ✓ 子系統二「巨量車載感測資料上傳系統」程式開發 (2016/12/20 ~ 2017/03/11)
  - ✓ 整合並進行系統測試 (2017/03/11 ~ 2017/05/31)
- 里程碑
  - ✓ 子系統一「巨量車載感測資料儲存系統」程式測試 (2017/12/21)
  - ✓ 子系統二「巨量車載感測資料上傳系統」程式測試 (2017/03/12)
  - ✓ 整合並進行系統測試 (2017/06/01)

#### 3.2 測試流程 (Testing Procedure)

##### 3.2.1 Subsystems Validation

本測試流程主要著重於第一年計畫各機制實作成果與系統整合測試－技術為基之智慧計算框架研發以支援先進駕駛輔助系統的主要成果項目包含下列所述。

1. 巨量車載感測資料儲存系統
2. 巨量車載感測資料上傳系統

##### 3.2.2 Integration Testing

本章節整合性測試，針對「技術為基之智慧計算框架研發以支援先進駕駛輔助系統」之各元件整合測試，並測試各子機制間相關的整合。

### 4. 測試結果與分析 (Test Result and Analysis)

子計畫二「技術為基之智慧計算框架研發以支援先進駕駛輔助系統之研究與實作」，目標即是處理並在雲端巨量資料儲存系統中處理與儲存具備巨量、高頻率等特性的車載感測資料，並且能維持雲端服務正常運作的系統架構。由於本整合型計畫的情境聚焦於先進駕駛輔助系統之雲端輔助設計優化，因此要在雲端的環境中處理大量的車載感測資料時，雲端服務必須有能夠處理巨量資料、高頻率資料及多樣性資料的需求。為了解決在雲端環境中，能夠處理具備上述三種巨量資料特性的車載感測資料所產生的問題，我們於子計畫二中建構出可提供對巨量車載感測資料進行高效率儲存的雲端儲存服務，並同時提供可以支援各分項計畫所需要之資料儲存的服務。我們的系統可以將不同車輛上的車載感測資料進行上傳，將資料上傳至我們所建構之巨量車載感測資料儲存系統上，以提供後續的資料分析及相關應用。



子計畫二在總計畫內提供功能包含：(1) 透過巨量車載感測資料上傳系統，將從許多車輛上所收集來包含有車載感測資料等資訊的檔案進行上傳，上傳到巨量車載感測資料儲存與分析平台上進行儲存。(2) 利用巨量資料儲存與分析平台將巨量的車載感測資料儲存至HDFS上，並利用Hadoop的巨量資料處理與分析的技術來對巨量車載感測資料進行分析。(3) 各分項計劃能透過通用型介面的方式 (RESTful API) 來使用巨量資料儲存平台，將所需進行儲存的資料儲存到HDFS。

本計畫分為三個子系統：

- 4.1 子系統一：巨量車載感測資料儲存與查詢系統
- 4.2 子系統二：巨量車載感測資料上傳系統

4.1 子系統一：巨量車載感測資料儲存與查詢系統

圖 2 為子系統一之系統架構圖。本子系統的主要目標是要提供多部車輛透過雲端儲存介面來存取大量的車輛感測資料 (OBD2、GPS 等)，以及透過 RESTful 的溝通方式讓各分項計畫可以對巨量車載感測資料儲存與分析平台進行車載感測資料或所需資料的存取。子系統一分別由兩個模組以及 Hadoop 巨量車載感測資料儲存與處理平台所組成，並敘述其功能如下：

- (1) Hadoop 巨量車載感測資料儲存與處理平台：此平台的主要目標是為了運用巨量資料處理平台的高可靠性、高效率和高擴充性的優勢，來擔任大規模車載感測資料儲存與檢索的核心平台，並且於平台上建置巨量車載感測資料儲存與查詢系統。
- (2) 檔案儲存模組：此模組主要是透過設定檔剖析模組所讀取之上傳設定，將多個車載感測資料檔案上傳至 HDFS 中進行儲存。
- (3) 資料查詢模組：子模組的主要功能是要提供各用戶可以透過查詢的服務請求，來對儲存於 HDFS 中的資料進行查詢。

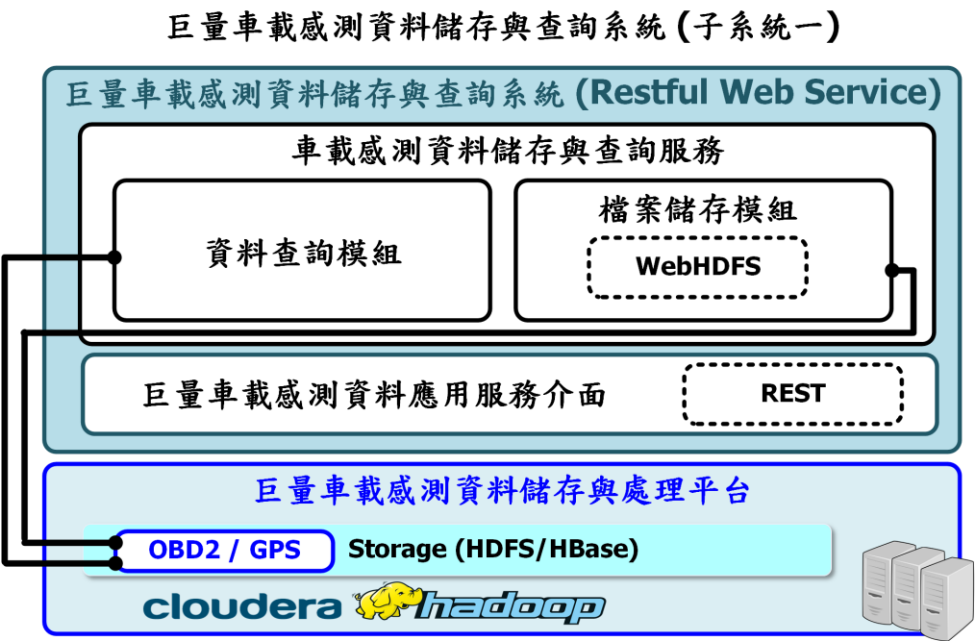


圖 2 子系統一之系統架構圖

子計畫二設計與規劃資料存取應用程式介面來達到標準化、共通性的資料存取規範，並開發 RESTful Web API 讓各子計畫能以 Http GET 和 POST 方式來存取巨量資料平台的 HDFS 資料。



圖 3 RESTful Web API 的 URL 結構

其中 Scheme 為 Http 型態；Host 與 Port 為伺服器服務的位置；ServiceRoot 為服務的名稱；ResourcePath 為 API 的路徑；QueryOptions 為查詢的條件。其中 QueryOptions 為根據表 2 的符號來組成查詢條件。

表 2 查詢條件之符號說明

符號	說明
?	資源路徑(resourcePath)和查詢項目(query options)之間的連結符號。
&	查詢項目若包含一個以上的查詢條件，則以& 連結。
=	查詢條件中參數名稱的參數輸入值。

在車載感測資料儲存與查詢服務中，主要提供了檔案儲存與資料查詢的功能，並且開發 RESTful 應用程式的介面，以提供各分項計畫的進行跨平台的資料溝通與資料傳遞。本子計畫提供六個應用程式介面，其說明如下表 3：

表 3 子計畫二提供的 API 功能列表

編號	事件類型	API名稱	功能說明
1	Event Sensor Data 事件感測資料	ESD_store	提供資料儲存功能，將ESD的JSON檔案儲存到HDFS。 (例如儲存ESD E01 20170530-123022.json)
2		ESD_retrieve	提供資料查詢功能，在HDFS取得ESD中指定的資料內容。 (例如取得OBD的資料內容)
3	Labeling Event Data 標記事件資料	LED_store	提供資料儲存功能，將LED的JSON檔案儲存到HDFS。 (例如儲存LED E01 20170530-123519.json)
4		LED_retrieve	提供資料查詢功能，在HDFS取得LED中指定的資料內容。 (例如取得演算法名稱)
5	Event Solution Model 事件解決模型	ESM_store	提供資料儲存功能，將ESM的JSON檔案儲存到HDFS。 (例如儲存ESD E01 20170530-123022.json)
6		ESM_retrieve	提供資料查詢功能，在HDFS取得ESM中指定的資料內容。 (例如取得演算法的參數)

接下來為巨量資料存取應用程式介面的結構與使用方式，我們分別敘述如下：

(1) **ESD\_store** (儲存事件感測資料)：提供資料儲存功能，將 ESD 的 JSON 檔案儲存到 HDFS。此 API 的輸入參數如下表 4 包含 CarID、EventID、Timestamp 以及 ESD 的 JSON 檔案；資料輸出則如下表 5，是以 JSON 格式回傳。

表 4 ESD\_store：輸入說明

參數名稱	資料型態	參數範例
CarID	string	CarID = 1HGCM82633A004352
EventID	string	EventID = E01
Timestamp	yyyy-MM-dd HH:mm:ss	Timestamp = 2017-05-30%2012:30:22 (%20為空白符號)
UploadFile	inputStream	ESD_E01_20170530-123022.json

表 5 ESD\_store：輸出說明

資料型態	資料輸出範例
JSON	檔案上傳成功的輸出內容： <pre>{ "results" : [ { "status" : "success" }, { "message" : "事件感測資料(Event Sensor Data)檔案上傳成功" } ] }</pre>
	檔案上傳失敗的輸出內容： <pre>{ "results" : [ { "status" : "false" }, { "message" : "事件感測資料(Event Sensor Data)檔案上傳失敗" } ] }</pre>

子系統一的測試情境為上傳車載感測資料，並驗證資料的可靠性與正確性。圖 4 為我們是使用 ESD\_store 來上傳車載感測資料，其 RESTful Web URL 為 [http://IP:8080/RESTful-Interface/ADAS-API-Service/ESD\\_store?CarID=1HGCM82633A004352&EventID=E01&Timestamp=2017-05-30%2012:30:22](http://IP:8080/RESTful-Interface/ADAS-API-Service/ESD_store?CarID=1HGCM82633A004352&EventID=E01&Timestamp=2017-05-30%2012:30:22)，其中條件的 CarID 為車輛編號，EventID 為事件編號，Timestamp 為資料上傳的時間，UploadFile 則由用戶端程式以 HttpPost 方式將檔案串流放置到 UploadFile 參數進行上傳。經由測試結果顯示，ESD\_store 回傳 {"results":[{"status":"success"}, {"message":"事件感測資料(EventSensorData)檔案上傳成功"}]}，代表資料上傳成功。

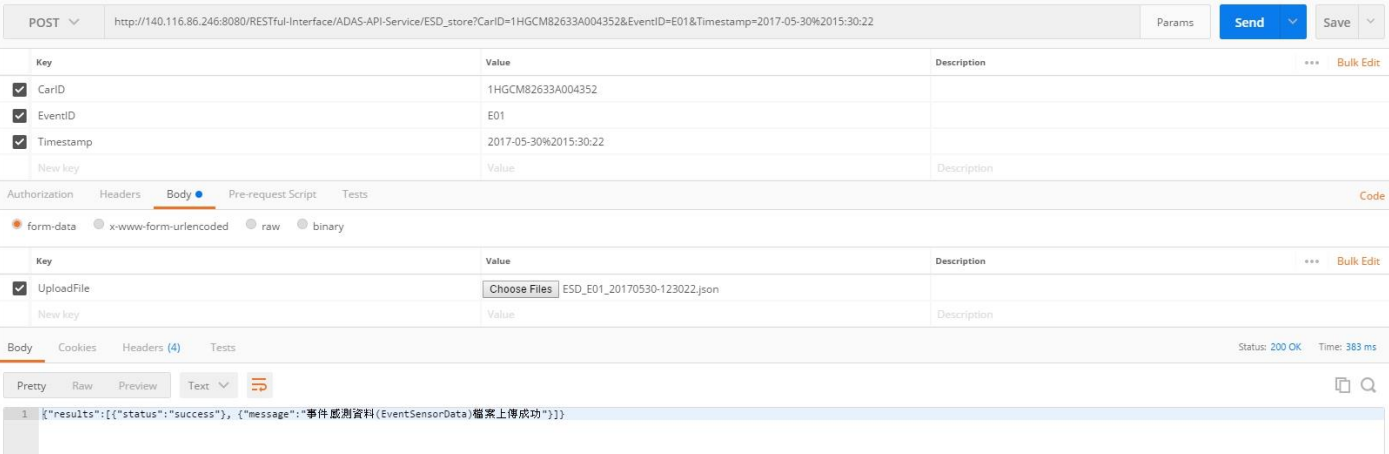


圖 4 ESD\_store：事件感測資料上傳畫面

(2) **ESD\_retrieve** (檢索事件感測資料)：提供資料查詢功能，在 HDFS 取得 ESD 中指定的資料內容。此 API 的輸入參數如下表 6，包含了 CarID、EventID、Timestamp 以及屬性名稱(AttrName)；資料輸出則如下表 7，是以 JSON 格式回傳。

表 6 ESD\_retrieve：輸入說明

參數名稱	資料型態	參數範例
CarID	string	CarID = 1HGCM82633A004352
EventID	string	EventID = E01
Timestamp	yyyy-MM-dd HH:mm:ss	Timestamp = 2017-05-30%2012:30:22 (%20為空白符號)
AttrName	JSONObject	例如檢索Engine RPM與Air Flow Rate的數值 AttrName={"AttrName":["Engine RPM","Air Flow Rate"] }

表 7 ESD\_retrieve：輸出說明

資料型態	資料輸出範例
JSON	Engine RPM與Air Flow Rate的數值輸出內容： { "results": [ { "Engine RPM": [762.5,766.1] }, { "Air Flow Rate": [2.45,2.48] } ] }

圖 5 為我們是使用 ESD\_retrieve 來檢索事件感測資料。其 RESTful Web URL 為 [http://IP:8080/RESTful-Interface/ADAS-API-Service/ESD\\_retrieve?CarID=1HGCM82633A004352&EventID=E01&Timestamp=2017-05-30%2012:02:45&AttrName={"AttrName":\["Vehicle\\_Identification\\_Number","Steering\\_Wheel","Image\\_Processing\\_Algorithms","Specifications"\]}](http://IP:8080/RESTful-Interface/ADAS-API-Service/ESD_retrieve?CarID=1HGCM82633A004352&EventID=E01&Timestamp=2017-05-30%2012:02:45&AttrName={)，其中條件的 CarID 為車輛編號，EventID 為事件編號，Timestamp 為資料上傳的時間，AttrName 則為欲進行檢索之屬性名稱。經由測試結果顯示，ESD\_retrieve 可以正確回傳欲進行檢索的結果。

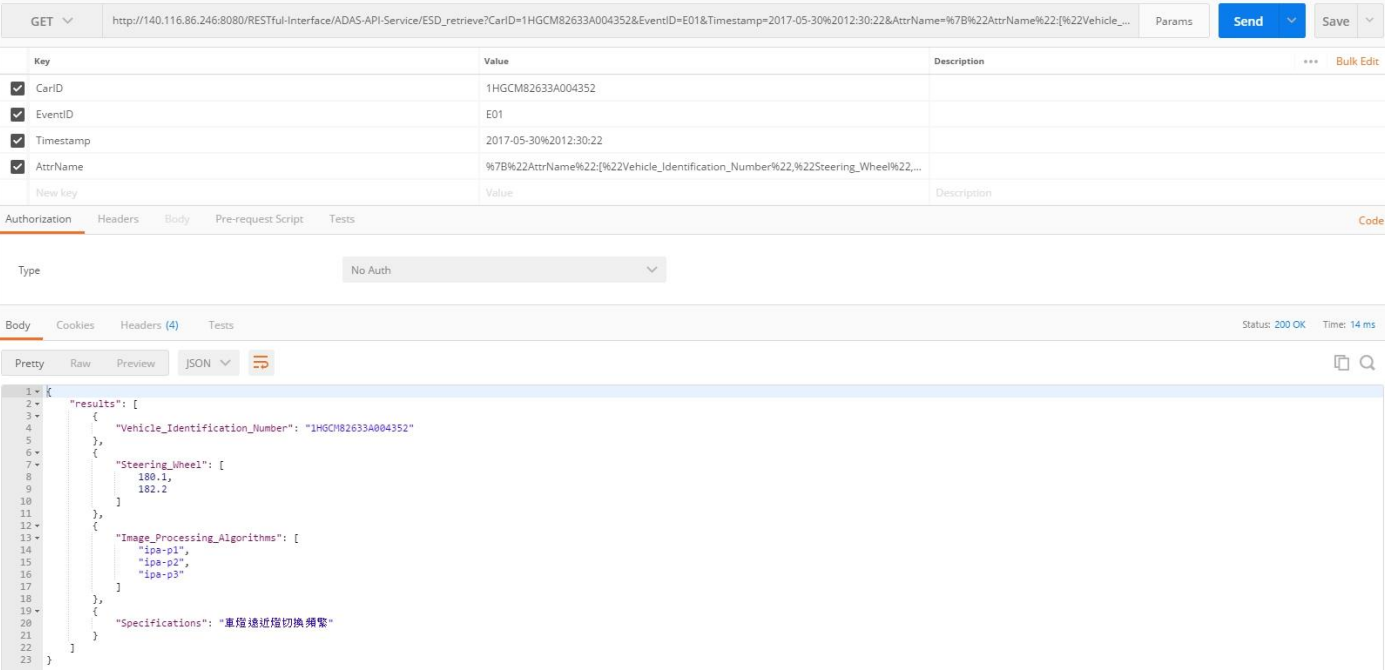


圖 5 ESD\_retrieve：事件感測資料檢索畫面

(3) **LED\_store** (儲存標記事件資料)：提供資料儲存功能，將 LED 的 JSON 檔案儲存到 HDFS。此 API 的輸入參數如下表 8，包含 CarID、EventID、Timestamp 以及 LED 的 JSON 檔案；資料輸出則如下表 9，以 JSON 格式回傳。

表 8 LED\_store：輸入說明

參數名稱	資料型態	參數範例
CarID	string	CarID = 1HGCM82633A004352
EventID	string	EventID = E01
Timestamp	yyyy-MM-dd HH:mm:ss	Timestamp = 2017-05-30%2012:35:19 (%20為空白符號)
UploadFile	inputStream	LED E01 20170530-123519.json

表 9 LED\_store：輸出說明

資料型態	資料輸出範例
JSON	檔案上傳成功的輸出內容： {" results" :[ {" status" : " success" }, {" message" : " 標記事件資料 (Labeling Event Data)檔案上傳成功" } ] }
	檔案上傳失敗的輸出內容：

	<pre>{   "results": [     {       "status": "false",       "message": "標記事件資料(Labeling Event Data)檔案上傳失敗"     }   ] }</pre>
--	---

圖 6 為我們是使用 LED\_store 來上傳標記事件資料，其 RESTful Web URL 為 [http://IP:8080/RESTful-Interface/ADAS-API-Service/LED\\_store?CarID=1HGCM82633A004352&EventID=E01&Timestamp=2017-05-30%2012:02:45](http://IP:8080/RESTful-Interface/ADAS-API-Service/LED_store?CarID=1HGCM82633A004352&EventID=E01&Timestamp=2017-05-30%2012:02:45)，其中條件的 CarID 為車輛編號，EventID 為事件編號，Timestamp 為資料上傳的時間，UploadFile 則由用戶端程式以 HttpPost 方式將檔案串流放置到 UploadFile 參數進行上傳。經由測試結果顯示，LED\_store 回傳{"results":[{"status":"success"}, {"message":"標記事件資料(LabelingEventData)檔案上傳成功"}]}，代表資料上傳成功。

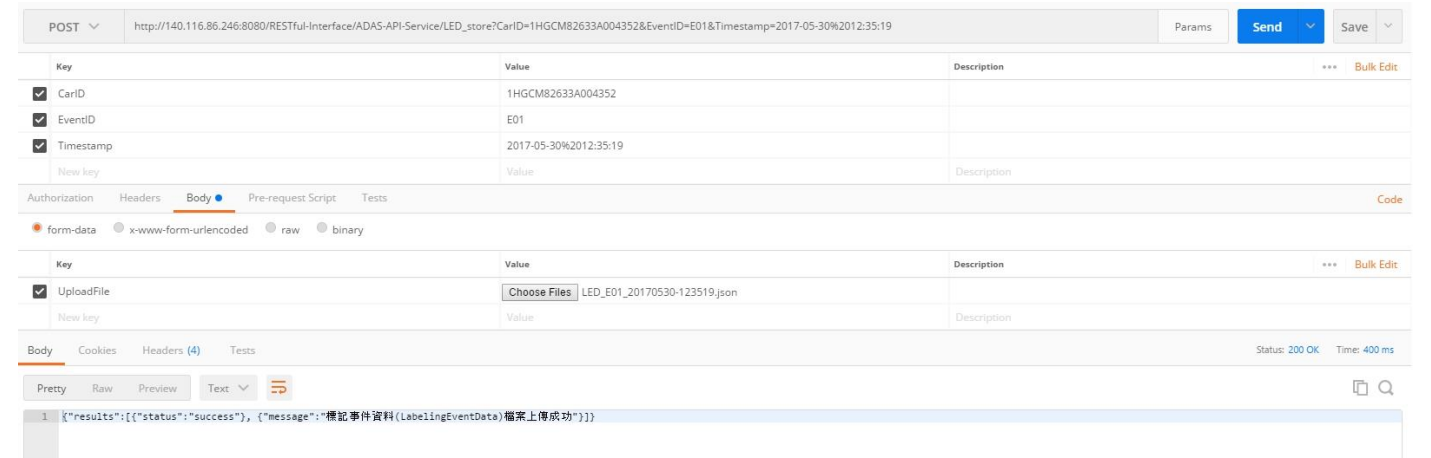


圖 6 LED\_store：標記事件資料上傳畫面

(4) **LED\_retrieve** (檢索標記事件資料)：提供資料查詢功能，在 HDFS 取得 LED 中指定的資料內容。此 API 的輸入參數包含 CarID、EventID、Timestamp 以及屬性名稱(AttrName)；資料輸出則以 JSON 格式回傳。

表 10 LED\_retrieve：輸入說明

參數名稱	資料型態	參數範例
CarID	string	CarID = 1HGCM82633A004352
EventID	string	EventID = E01
Timestamp	yyyy-MM-dd HH:mm:ss	Timestamp = 2017-05-30%2012:35:19 (%20為空白符號)
AttrName	JSONObject	例如檢索Road Detection Model的參數標記 AttrName={"AttrName":["Road Detection Model"] }

表 11 LED\_retrieve：輸出說明



資料型態	資料輸出範例
JSON	Road Detection Model的參數標記輸出內容： <pre> {   "results": [     {       "Road Detection Model": {         "P1": {           "Kernel Size": "true",           "Cell Size": "true",           "Block Size": "true",           "Threshold": "true"         },         "P2": "false"       }     }   ] }</pre>

圖 7 為我們是使用 LED\_retrieve 來檢索標記事件資料。其 RESTful Web URL 為 [http://IP:8080/RESTful-Interface/ADAS-API-Service/LED\\_retrieve?CarID=1HGCM82633A004352&EventID=E01&Timestamp=2017-05-30%2012:35:19&AttrName={\"AttrName\":\[\"Vehicle\\_Identification\\_Number\",\"Road\\_Detection\\_Model\",\"Front\\_Object\\_Detection\\_Model\",\"Event\\_Description\"\]}](http://IP:8080/RESTful-Interface/ADAS-API-Service/LED_retrieve?CarID=1HGCM82633A004352&EventID=E01&Timestamp=2017-05-30%2012:35:19&AttrName={\)，其中條件的 CarID 為車輛編號，EventID 為事件編號，Timestamp 為資料上傳的時間，AttrName 則為欲進行檢索之資料屬性名稱。經由測試結果顯示，LED\_retrieve 回傳欲進行檢索的結果如下。

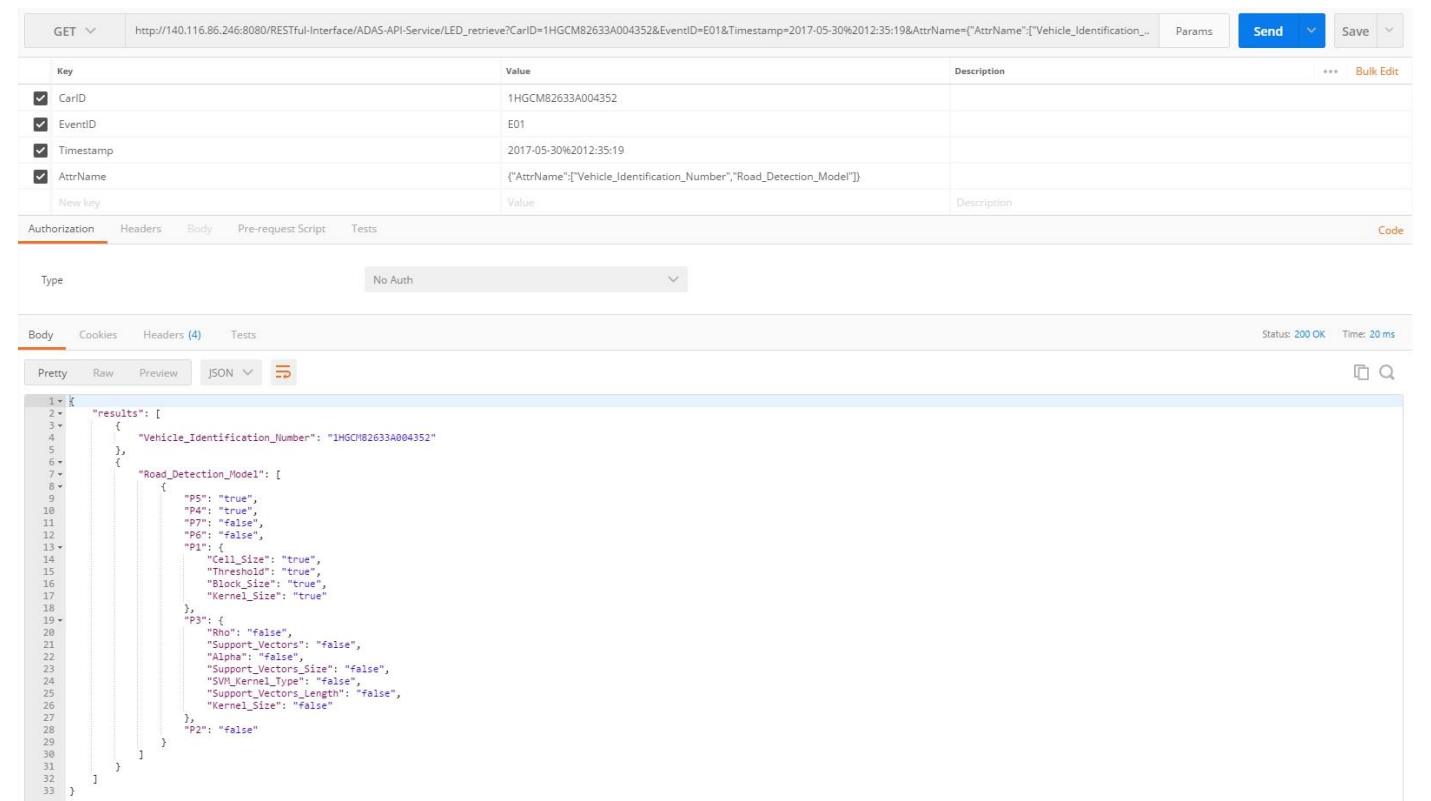


圖 7 LED\_retrieve：標記事件資料檢索畫面

(5) **ESM\_store** (儲存事件解決模型)：提供資料儲存功能，將 ESM 的 JSON 檔案儲存到 HDFS。此 API 的輸入參數如下表 12，包含有 CarID、EventID、Timestamp 以及 ESM 的 JSON 檔案；資料輸出則如下表 13，是以 JSON 格式回傳。

表 12 ESM\_store：輸入說明

參數名稱	資料型態	參數範例
CarID	string	CarID = 1HGCM82633A004352
EventID	string	EventID = E01
Timestamp	yyyy-MM-dd HH:mm:ss	Timestamp = 2017-05-30%2015:31:45 (%20為空白符號)
UploadFile	inputStream	ESM E01 20170530-153145.json

表 13 ESM\_store：輸出說明

資料型態	資料輸出範例
JSON	檔案上傳成功的輸出內容： <pre>{" results" :[ {" status" : " success"  }, {" message" : " 事件解決模型(Event Solution Model)檔案上傳成功"  } ] }</pre>
	檔案上傳失敗的輸出內容： <pre>{" results" :[ {" status" : " false"  }, {" message" : " 事件解決模型(Event Solution Model)檔案上傳失敗"  } ] }</pre>

圖 8 為我們是使用 ESM\_store 來上傳事件解決模型，其 RESTful Web URL 為 [http://IP:8080/RESTful-Interface/ADAS-API-Service/ESM\\_store?CarID=1HGCM82633A004352&EventID=E01&Timestamp=2017-05-30%2012:02:45](http://IP:8080/RESTful-Interface/ADAS-API-Service/ESM_store?CarID=1HGCM82633A004352&EventID=E01&Timestamp=2017-05-30%2012:02:45)，其中條件的 CarID 為車輛編號，EventID 為事件編號，Timestamp 為資料上傳的時間，UploadFile 則由用戶端程式以 HttpPost 方式將檔案串流放置到 UploadFile 參數進行上傳。經由測試結果顯示，ESM\_store 回傳{"results":[{"status":"success"}, {"message":"事件解決模型(EventSolutionModel)檔案上傳成功"}]}，代表資料上傳成功。



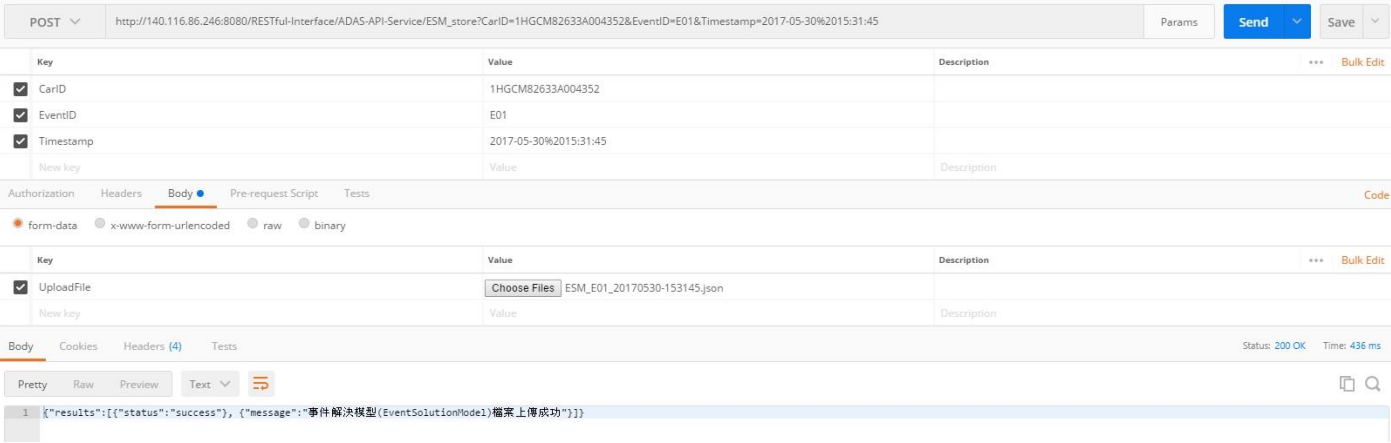


圖 8 ESM\_store：事件解決模型上傳畫面

(6) **ESM\_retrieve** (檢索事件解決模型)：提供資料查詢功能，在 HDFS 取得 ESM 中指定的資料內容。此 API 的輸入參數如下表 14，包含 CarID、EventID、Timestamp 以及屬性名稱(AttrName)；資料輸出則如下表 15，是以 JSON 格式進行回傳。

表 14 ESM\_retrieve：輸入說明

參數名稱	資料型態	參數範例
CarID	string	CarID = 1HGCM82633A004352
EventID	string	EventID = E01
Timestamp	yyyy-MM-dd HH:mm:ss	Timestamp = 2017-05-30%2015:31:45 (%20為空白符號)
AttrName	JSONObject	例如檢索Description和Intelligent Headlight Model的參數設定值 AttrName={"AttrName":["Description","Intelligent Headlight Model"]  }

表 15 ESM\_retrieve：輸出說明

資料型態	資料輸出範例
JSON	Description和Intelligent Headlight Model的參數設定值輸出內容： {" results" :[ {" Solution Description" :[ {" Description" :[" 輪胎需要打氣" ," 後視鏡需要更換" ] } ],  "Intelligent Headlight Model":[ { "P1": "74" } , { "P2": "226" } ] } ] }

圖 9 為我們是使用 ESM\_retrieve 來檢索事件解決模型。其 RESTful Web URL 為 [http://IP:8080/RESTful-Interface/ADAS-API-Service/ESM\\_retrieve?CarID=1HGCM82633A004352&Even](http://IP:8080/RESTful-Interface/ADAS-API-Service/ESM_retrieve?CarID=1HGCM82633A004352&Even)

tID=E01&Timestamp=2017-05-30%2015:31:45&AttrName={"AttrName":["Vehicle\_Identification\_Numbe  
r","Description","Road\_Detection\_Model"]}

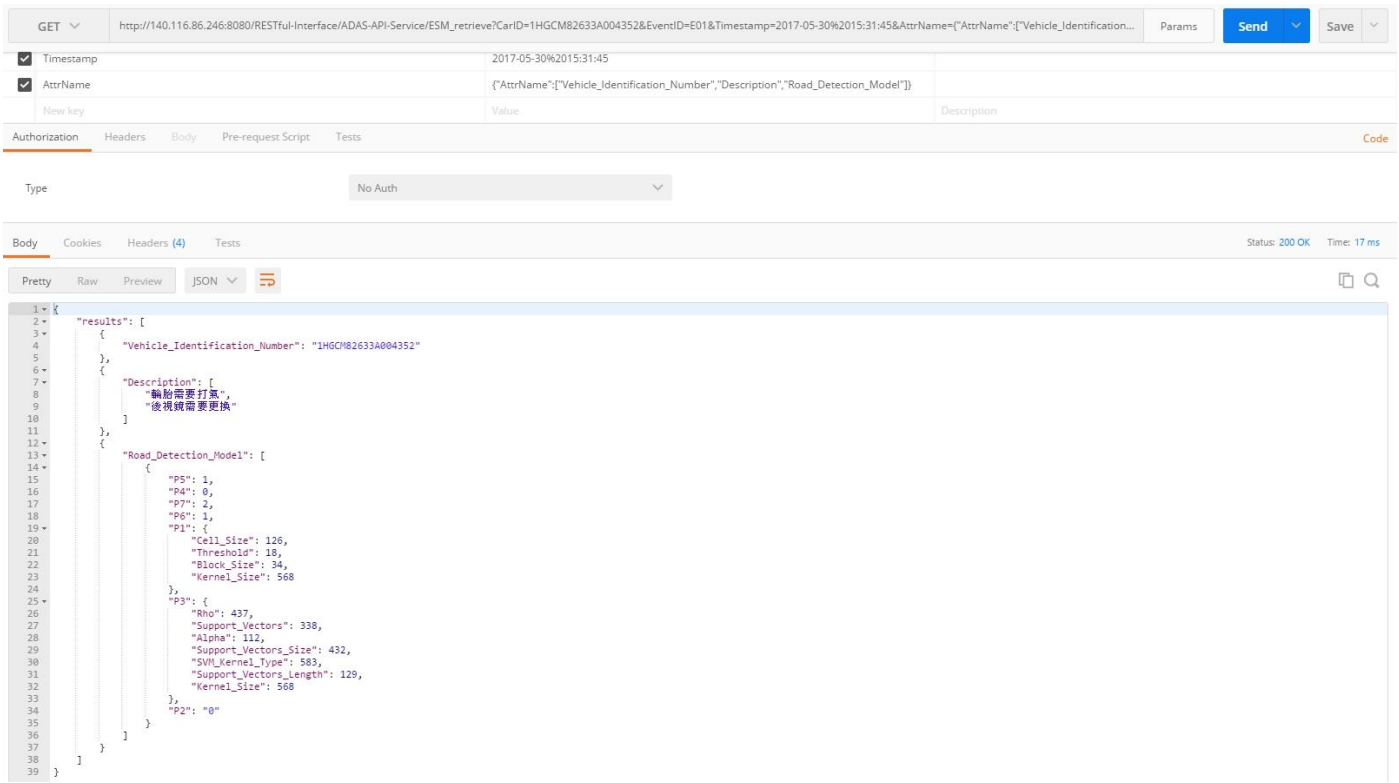
，其中條件的 CarID 為車輛編號，EventID 為事件編號，Timestamp 為資料上傳的時間，AttrName 則為欲進行檢索之資料屬性名稱。經由測試結果顯示，ESM\_retrieve 回傳欲進行檢索的結果如下。

圖 9 ESM\_retrieve：事件解決模型檢索畫面

4.2 子系統二：巨量車載感測資料上傳系統

本子系統的目的是將不同車輛上所收集到的車載感測資料上傳到目標 HDFS。圖 10 為巨量車載感測資料上傳系統架構圖，分別由兩個模組組成，並敘述其功能如下：

- (4) 檔案讀取模組：此模組的主要功能是要讀取多個車載感測資料的檔案，並將其進行壓縮，以節省網路傳輸所使用的頻寬，以及上傳所消耗的時間。
- (5) 設定檔剖析模組：此模組的功能是能透過剖析容易進行修改之 JSON 檔案，利用 JSON 檔中所含有的上傳目標位置等資訊進行模組上傳功能的設定。
- (6) 檔案上傳模組：此模組主要是透過設定檔剖析模組所讀取之上傳設定，將多個車載感測資料檔案上傳至 HDFS 中進行儲存。

## 巨量車載感測資料上傳系統 (子系統二)

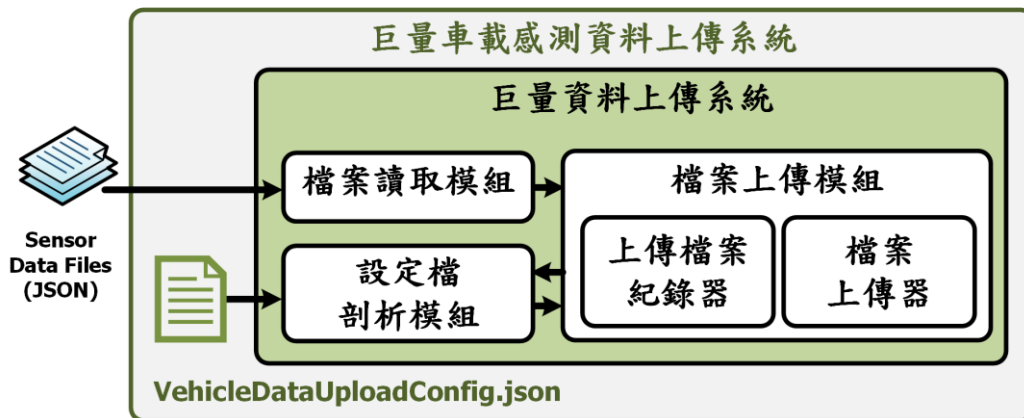


圖 10 子計劃二的子系統架構圖

本子系統二的測試情境為上傳 250 個車載感測資料檔案至 HDFS 中進行儲存，並驗證資料儲存的可靠性以及正確性。首先，圖 11 為我們實際收集了 250 個車載感測資料，我們所收集的車載感測資料包含有：油箱燃油狀況、引擎附載量、車輛引擎轉速、車輛行駛時速、引擎進氣溫度、引擎冷卻水溫、車輛位置的經緯度等三十七項。接著我們透過巨量車載感測資料上傳系統進行資料上傳，其中圖 12 為此子系統的上傳設定檔，本設定檔使用陣列可以達到資料維護方便、可閱讀性高的優勢，並且可以根據應用情境的需求來進行欄位內容與數量的調整。此設定檔中的資訊包含了：巨量資料處理平台的伺服器位置、巨量資料處理平台的伺服器埠、車輛感測資料的存放目錄路徑、資料上傳紀錄日誌 (log) 的存放目錄路徑、資料上傳速率 (秒)。最後子系統二會根據此設定檔將 250 個檔案上傳至 HDFS 中，圖 13 為利用 Hue 檢視於 HDFS 上的車輛感測資料檔案，顯示子系統二能將所有的檔案都上傳至 HDFS 當中。圖 14 為利用 Hue 檢視於 HDFS 上的各車輛感測資料檔案的資料內容，其中紀錄了該檔案中的三十七項感測資料。最後，子系統二會紀錄上傳成功與失敗的車輛感測資料的檔案名稱以及上傳時間，圖 15 為上傳成功的紀錄，共有 250 筆。

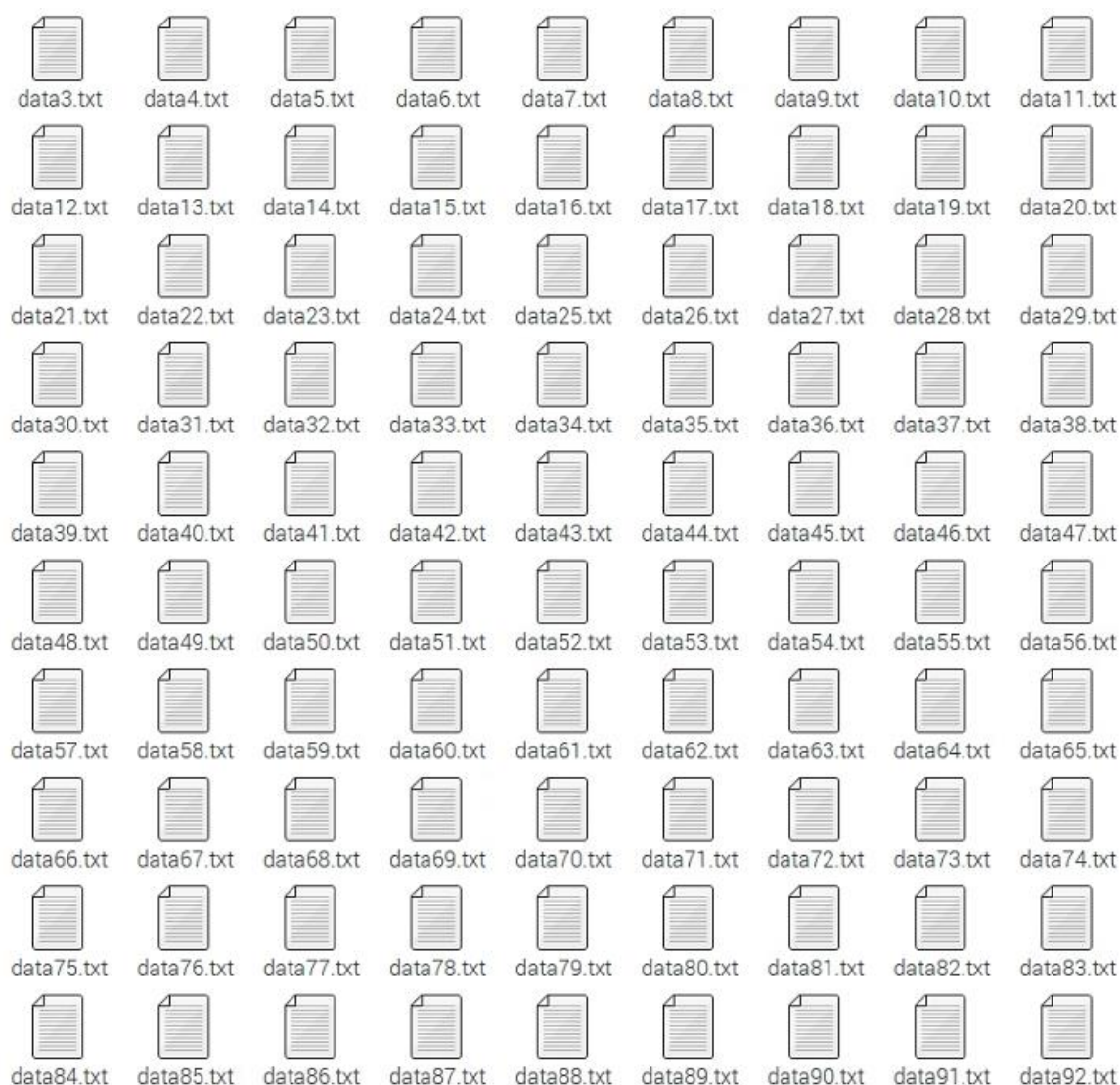
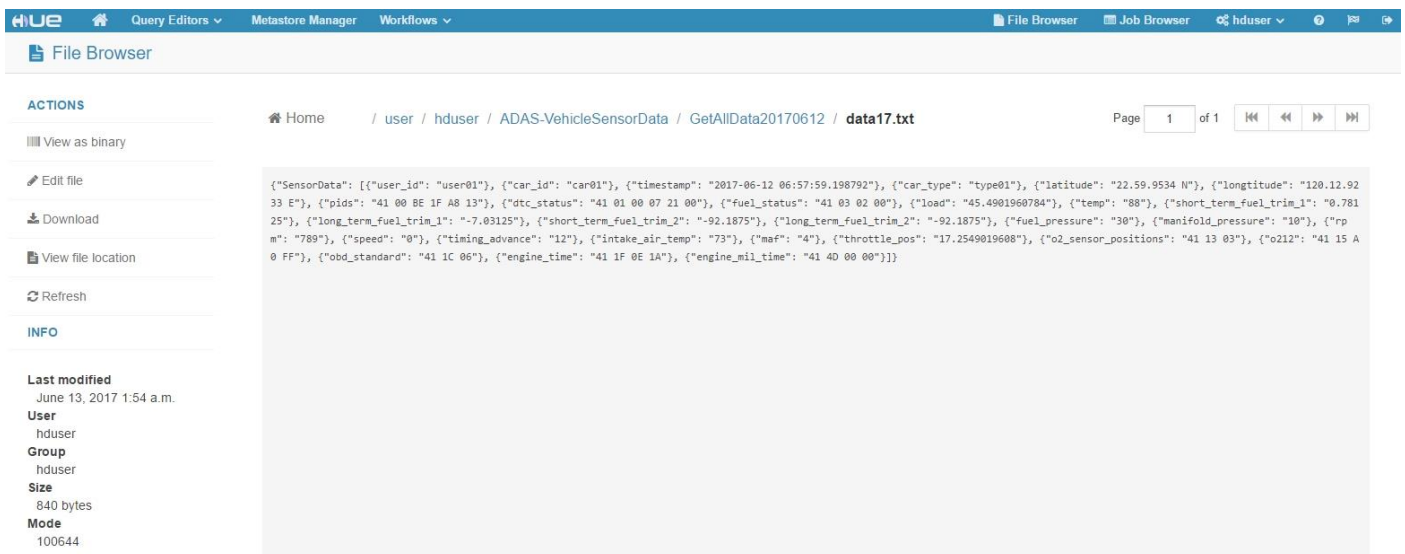
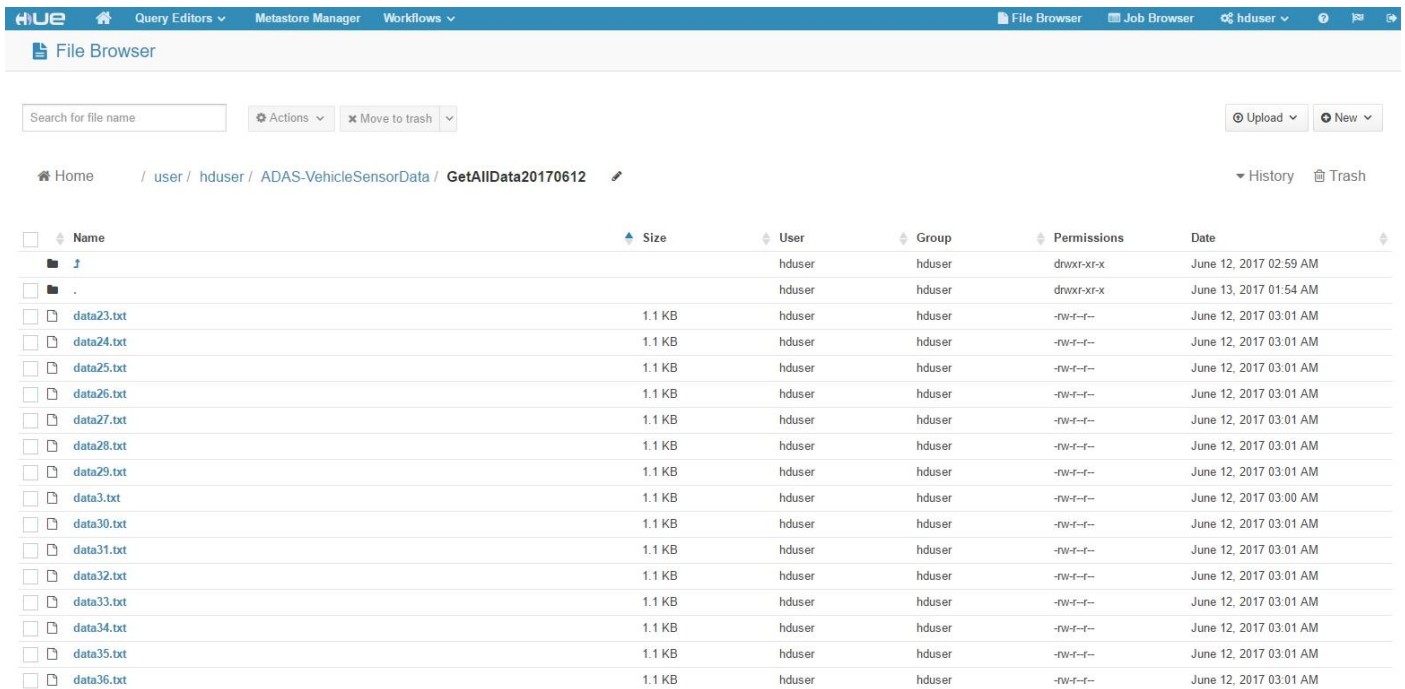


圖 11 實際收集的車載感測資料檔案 (共 250 個)

```
{
  "target_Select": "Target_1",
  "Target_1": [
    {
      "IP": "██████████",
      "Port": "8080",
      "file_Location": "/home/pi/file/",
      "log_Location": "/home/pi/file/header/",
      "upload_Rate": "0"
    }
  ]
}
```

圖 12 巨量車載感測資料上傳系統：車輛資料上傳設定檔





```
2017/06/12 05:14:28 data80.zip
2017/06/12 05:14:28 data206.zip
2017/06/12 05:14:29 data205.zip
2017/06/12 05:14:30 data141.zip
2017/06/12 05:14:30 data51.zip
2017/06/12 05:14:31 data135.zip
2017/06/12 05:14:32 data5.zip
2017/06/12 05:14:33 data10.zip
2017/06/12 05:14:34 data76.zip
2017/06/12 05:14:35 data128.zip
2017/06/12 05:14:36 data163.zip
2017/06/12 05:14:37 data14.zip
2017/06/12 05:14:38 data145.zip
2017/06/12 05:14:39 data74.zip
2017/06/12 05:14:40 data165.zip
2017/06/12 05:14:40 data168.zip
2017/06/12 05:14:41 data93.zip
2017/06/12 05:14:42 data208.zip
2017/06/12 05:14:43 data182.zip
2017/06/12 05:14:44 data114.zip
2017/06/12 05:14:45 data87.zip
2017/06/12 05:14:45 data152.zip
2017/06/12 05:14:47 data45.zip
2017/06/12 05:14:49 data129.zip
2017/06/12 05:14:50 data189.zip
2017/06/12 05:14:50 data66.zip
2017/06/12 05:14:51 data146.zip
2017/06/12 05:14:52 data217.zip
2017/06/12 05:14:53 data110.zip
2017/06/12 05:14:53 data193.zip
2017/06/12 05:14:54 data197.zip
2017/06/12 05:14:55 data32.zip
```

圖 15 巨量車載感測資料上傳系統：上傳成功紀錄檔之內容

子系統二經整合與測試的結果顯示，能有效將 250 個車載感測資料上傳至 HDFS 中進行儲存，並花費大約 40 秒鐘的時間。測試實驗結果可以將大量的車載感測資料正確地上傳至 HDFS 中。

### 4.3 計畫測試結果

如表所示，依據上述的子系統並進行實際系統測試，其結果如下所示。

表 16 計畫測試結果

Test Case	Result(Pass / Fail)	Comment
子系統一	Pass	符合測試狀況。
子系統二	Pass	符合測試狀況。
<b>RATE</b>	<b>100 %</b>	