big data and cognitive computing

Article

Big Data and Personalisation for Non-Intrusive Smart Home Automation

Suriya Priya R. Asaithambi [1], Sitalakshmi Venkatraman [2,*] and Ramanathan Venkatraman [1]

[1] Institute of Systems Science, National University of Singapore, Singapore 119077, Singapore; suria@nus.edu.sg (S.P.R.A.); rvenkat@nus.edu.sg (R.V.)
[2] School of Engineering, Construction & Design, Melbourne Polytechnic, Preston, VIC 3072, Australia
* Correspondence: sitavenkat@melbournepolytechnic.edu.au; Tel.: +61-3-9269-1711

Abstract: With the advent of the Internet of Things (IoT), many different smart home technologies are commercially available. However, the adoption of such technologies is slow as many of them are not cost-effective and focus on specific functions such as energy efficiency. Recently, IoT devices and sensors have been designed to enhance the quality of personal life by having the capability to generate continuous data streams that can be used to monitor and make inferences by the user. While smart home devices connect to the home Wi-Fi network, there are still compatibility issues between devices from different manufacturers. Smart devices get even smarter when they can communicate with and control each other. The information collected by one device can be shared with others for achieving an enhanced automation of their operations. This paper proposes a non-intrusive approach of integrating and collecting data from open standard IoT devices for personalised smart home automation using big data analytics and machine learning. We demonstrate the implementation of our proposed novel technology instantiation approach for achieving non-intrusive IoT based big data analytics with a use case of a smart home environment. We employ open-source frameworks such as Apache Spark, Apache NiFi and FB-Prophet along with popular vendor tech-stacks such as Azure and DataBricks.

Keywords: smart home; internet of things; IoT; home automation; big data; machine learning

Citation: Asaithambi, S.P.R.; Venkatraman, S.; Venkatraman, R. Big Data and Personalisation for Non-Intrusive Smart Home Automation. *Big Data Cogn. Comput.* **2021**, *5*, 6. https://doi.org/10.3390/bdcc5010006

Received: 30 November 2020
Accepted: 26 January 2021
Published: 30 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the last decade, home appliances have transitioned from being made up of simple devices supported by low-cost sensors to intelligent devices that can detect human movement [1–3]. A smart home is an environment where household appliances can be monitored and controlled remotely based on the state of various built-in devices such as sensors and actuators [4]. More commonly, a smart home has many standalone appliances with a convenient way to automatically identify, monitor and control an appliance to perform various operations such as change its on/off status [5]. This is normally possible due to seamless integration of technologies via the Internet and the advancements in Internet of Things (IoT) [6,7].

The current smart home automation is based on a rules-engine for making decisions as defined by the user of the system. These automated decisions are usually guided by a set of rules defined by the user in a computational element, which has the information received from the sensors data [8]. However, these decision rules do not work due to the changing external environment or the nature of the things/device in the network [9,10]. For instance, based on the external factors affecting the room temperature, an air-conditioner in the room could be automatically set to on/off operational status according to the user's preference while present in the room. The current smart home systems can do presence detection only based on motion sensor data [11,12]. Such passive motion sensors based on infrared and other similar technologies have the disadvantage of failing to detect objects when the user is not in motion for a long time [13]. Another alternative to overcome this drawback is to employ vision-based, wearable, or other similar ubiquitous technologies for continuous object detection [4]. Such active sensors collect more expressive information including users' private data, and it is a challenge to extract anonymous information with correctness while trying to enforce privacy [8,10]. In summary, there are two main reasons preventing adoption of such active and invasive sensing technologies in smart homes: (i) difficult to generalise models with changing environment conditions, and (ii) user concerns with perceived intrusiveness. The term "intrusiveness" can be related to many aspects, such as: (i) sensing technologies associated with sensors requiring user-based inputs (e.g., touch-based or user controls certain parameters), (ii) monitoring methods that can interfere with the system processing or user privacy (e.g., video cameras), (iii) data collection and storage methods having an impact on data security and user privacy (e.g., raw personal data not anonymised/encrypted/authenticated access). The behaviour of the sensors cum actuators (intrusive or non-intrusive) can vary significantly depending on the building context. It is feasible to achieve accurate fine-grained occupancy counting using information available from non-intrusive data sources like room temperature, cooling devices, water consumption and a number of other Wi-Fi connected devices. Overall, intrusiveness is perceived by users based on the level of personal data collection and/or user's operational interference. The concept of intrusiveness is also very much associated with security and privacy in smart homes due to factors such as occupancy surveillance and secondary use of personal information. Recognising human activities has been used effectively for human detection and personalisation in an IoT environment [7]. Personalisation can be achieved with various levels of intrusiveness of operational interference, user or system interventions and user privacy and security concerns. The challenge is to achieve personalisation in the context of home automation with the least intrusiveness. In this paper, we focus on developing a non-intrusive smart home automation system with a novel IoT technology instantiation approach by integrating big data collected from Wi-Fi sensors in a smart home and combining these with spatial activities for deriving user personalisation.

In a typical home automation environment, the most predominantly employed sensors are passive in operation as they can trigger events only when there is an action performed [13]. Passive sensors are more preferred as they are operated by battery and can be easily relocated and scaled based on user requirements. Such sensors have been operational for several years. However, they lack the essential intelligent computations for a smart home environment. By leveraging on rich data exchange among the sensors in an IoT network, the decision rules defined within the computational elements can be employed to intelligently operate various other essentials of the home environment resulting in a smart home automation [14]. Hence, in this paper, apart from user presence detection, we explore integration of intelligent computation of user's preferences based on historical big data collected in an IoT environment using a non-intrusive modelling approach. By incorporating personalisation based on any user's behaviour patterns in an IoT environment, our proposed system model can provide a smarter response without requiring any user intervention achieving a non-intrusive smart home automation.

Most smart home initiatives developed in the past decades have adopted models that are not popular due to factors such as high cost, difficulty in installation, technology limitations, highly intrusive and lack of data accuracy and intelligent management services [2]. Recent advancements in IoT offer new opportunities to facilitate seamless Wi-Fi based sensor connectivity of low-cost devices and collection of large volume of data sensor networks that are key features in developing an intelligent home environment. However, due to pitfalls in integration and lack of non-intrusiveness in the collection and processing of sensor data, there are still concerns in realising a viable smart home automation of the future [15,16]. The majority of existing studies have explored intertwining only a couple of domains, and predominantly in other problem domains with the least application to smart homes [17,18]. Further, only recently personalisation and non-intrusiveness have been given much importance with more IoT technologies supporting Ambient Assistant Living (AAL) using behaviour monitoring through non-invasive and privacy preserving sensing in many developed countries [6,14]. Several use cases are being developed in the AAL domain and this paper aims to advance further in this research direction. Our aim in this paper is to offer a novel technology instantiation and live data stream integration approach to a typical home automation model and demonstrate the intersection of four key domain areas, namely, non-intrusive sensor technologies, IoT, Big Data, and machine learning (ML) by implementing a prototype as a use case. We propose an approach to integrate and personalise data from user activities and real-time data streams of the IoT devices in a home environment using ML algorithms within a big data architecture. To achieve this, our model for a smart home automation system makes use of primarily the four v's (variety, veracity, velocity, and value) of big data from the IoT ecosystem of a home environment. For timeseries forecasting at a scale, we employ ML models such as modular regression and FBProphet as an open-source software tool to detect and estimate user occupancy, activities and other behaviour patterns non-intrusively. For occupancy analysis we used a logistic regression model.

Overall, this paper advances research by taking initial steps in achieving a personalised smart home automation system. The contribution of our work to the academic and practical value are threefold: (i) proposal of a novel technology instantiation approach for a typical smart home automation model, (ii) development of a non-intrusive approach of collecting and integrating sensor data from various IoT devices to perform ML based analysis of user-centric patterns for intelligent decision making, and (iii) implementation of the model by integrating cloud and big data analytics solutions using open-source technology standards to provide a use case of the application of our model for a practical smart home environment.

The rest of the paper consists of the following sections. In Section 2, we summarise our literature survey on related work and the need for the focus towards non-intrusive home automation. Section 3 describes our proposed novel non-intrusive modelling approach using big data and personalisation to achieve smart home automation. In Section 4, we present the use case of our modelling approach to develop a prototype as a demonstration of our implementation strategy. Section 5 provides a comprehensive discussion on the design and implementation considerations as well as limitations resolved in this work. Finally, Section 6 concludes with future research directions.

2. Background and Need for the Study

Research in home appliance technologies over the past two decades were more focused on developing new electrical gadgets to achieve efficiency through smart use of electricity [18,19]. Studies have reported the technological advancements in the smart home arena coming from three generations based on electromagnetic contacts, temperature sensors, smart power accessories, passive RFID and infrared based sensors. Some of these sensors provide very little data, and further, low-cost devices are prone to data noise and computational limitations. While wearable devices such as personal health tracking gadgets can provide more accurate and rich data, they are intrusive and need to be charged and always operational [19]. With the recent advancement in the IoT landscape, various web-based applications could be used to distribute and process information via the Internet and the wireless home network connected to different home appliances and sensors. Recently, the term 'smart home' has been more interchangeably used with 'home automation' with an aim to enhance quality of life and personal wellbeing using automatic voice detection [20–22]. Some home automation systems have been developed with specific objectives such as improving energy-efficiency, safety and home protection from intruders. There is a need for load balancing of energy to allow several devices to be operational simultaneously and to be controlled by different users in the home with an easy-to-use interface that can even cater to multiple home occupants' preferences [23]. While vision-based technologies making use of artificial intelligence are sophisticated, they are perceived intrusive and their automation models cannot be generalised or integrated with other technologies [24]. Further, environmental conditions such as light, heat and other changing conditions that affect the model could be quite challenging to be incorporated in a smart home automation system. User-friendly smart phones are also being employed for personalisation and automation control [25,26]. However, for future sustainable smart home automation, we need to make use of the hidden potential of big data captured in the smart home ecosystem. Artificial intelligence techniques such as machine learning for big data analysis and decision making could be employed in a non-intrusive manner to achieve a personalised smart home automation [27–29].

Some studies have explored certain non-intrusive data collected from sensors such as user's gait and weight to identify the user and achieve personalisation to a certain extent in a restricted environment [30]. Certain other studies have further captured environmental information, and by exchanging data among sensors and users via the ubiquitous home network have modelled a smart home environment to operate without the need for user intervention [9,22,31]. A recent study has demonstrated personalisation by making dynamic modifications to the applications and interfaces of a home environment based on the changing user needs in a smart home environment [1]. However, such existing proposals in literature lack a big data management approach as the decisions made are not based on intelligent processing of big data from various sources. It is important to integrate non-intrusive user behaviour patterns with sensor data stored as big data sources to achieve a more accurate personalisation. Another work proposed a federation of IoT testbeds to provide an experimental infrastructure that forms a semantically enabled multi-domain data marketplace that can be accessed on-line [32]. Though they serve as a big data infrastructure, they lack intelligent data processing capabilities.

Previously, a generic big data architecture was proposed using open-source tools to effectively analyse a huge amount of live healthcare data streams using distributed real-time computing, and a distributed storage system [33]. The big data architecture was modelled to process the healthcare data streams for real-time analytics and this restricts its application for a smart home automation system. Since big data streams from several IoT devices and appliances are generated every day in a smart home environment, there is a need for modelling big data along with user personalisation for making autonomous decisions in real-time. This is possible by employing intelligent computations and ML.

Overall, there is a great need to integrate IoT technologies with big data and smart home automation requires ML for non-intrusive personalisation to achieve a sustainable solution that is flexible and adaptive over time. Existing research shows growing interest to recognise human activity and identify methods related to context modelling, reasoning and management. The research emphasis in such studies pertains to Context and Activity Modeling and Recognition (CoMoRea). However, the existing literature has a scarcity of exploration into technology integration, which is crucial for a successful smart home automation system deployment. This gap in technology integration literature forms the key motivation for our proposed approach. Our research prototype is an attempt to integrate the disparate data collected from cloud connected smart home controllers into a big data lake and process. Further, we consider known user behaviour patterns with several IoT sensor data in a smart home environment. Our proposed deployment approach study effectively intersects four areas that are currently wide research interest such as passive sensor technologies, IoT, big data and ML for achieving non-intrusive smart home automation.

3. Modelling Big Data and Personalisation for Smart Home Automation

Identifying the real-time human behaviour patterns associated in a smart home is an opportunity for modelling personalisation [14,23]. This can be achieved via user profiling of real-time data generated from the devices and sensors operated by the user [34]. In addition, various data elements and attributes from different environmental sources can be analysed to understand data patterns that depict a particular user's behaviour and physical activities providing a personalised profiling for the user in the smart home. The computational element within the environment has the potential to personalise the smart home environment if it can identify the user within the environment. Such identification is currently possible with obtrusive sensors like cameras, microphones, and body wearables. To a large extent, many users tend to avoid intrusive sensors deployed in their home environment due to privacy concerns and prefer a privacy-enabled system with smart home automation [14,26,34].

Our proposed model leverages on non-intrusive sensors such as motion sensors, door contact sensors, temperature and humidity sensors, pressure sensors, fire alarms and smoke sensors. These are typical sensors that are usually deployed in a smart home environment and are well accepted by users as they are not intrusive of user privacy unlike those IoT devices that can capture user activity using video or voice data formats [22,35]. In addition to identifying a user within a smart home, the personalisation profiling component is modelled to automatically control the home environment intelligently based on each user preference and a big data stream of activities.

By carefully choosing devices with appropriate sensors and actuators to communicate bidirectionally with smart home controllers, smart home devices can communicate and control each other more effectively. We extend this generally adopted peer-to-peer architecture further by leveraging on recent advancements in the scalable computations and storage facilities of cloud and big data processing techniques. From current state-of-the-art IoT capabilities of smartness (quirky intelligence), the aim of the proposed model is towards achieving a better optimal, personalised, long term and balanced intelligent decision in a smart home environment.

A generic IoT driven architecture that extends cloud and big data processing abilities typically consists of five component layers. Figure 1 illustrates the component layering of the IoT driven architecture adopted for our modelling. An overview of the five component layers are given below:

(1) Sensors—these are used to mainly collect and transduce the data.
(2) Computing—this is usually a cloud service to process the big data received from plenty of sensors using ML and various other computational models of data analytic. This layer can extend any big data processing or cloud native capabilities as demanded by the device automation.
(3) Collectors—these facilitate in the collection of messages sent by the computing algorithms or associated devices as decision or action instructions.
(4) Actuators—they use the information received from the sensor and/or from the Internet, and then trigger the associated device to perform a particular function based on the decisions taken by the computing node and big data processing functions.
(5) (Client) Device—this performs the desired interaction with smart devices on behalf of the user when triggered. In an IoT landscape, the sensors, actuators, collectors, communication gateways, etc., are often depicted as smart devices whereas the ones interacting with the user are considered as edge devices.

With the recent advancement in wireless protocols and HTTP based protocols, both sensors and actuators are present in most home appliances. Also, the device connectivity and control spectrum are increased by smart home controllers such as Amazon Echo and Google Home. Human interfacing happens via smartphone apps that are considered as end devices. Our proposal focusses on the ability to connect people and things in a personalised manner, thereby achieving a novel capability of collaboration, control, measurements, inferences, and changes to a smart home ecosystem.

Our proposed model for smart home automation is based on massive non-intrusive data collected with much importance given to the following four v's of Big Data, namely variety, veracity, velocity and value, which can cater to the main objective of achieving automatic personalised decision making:

(i) Big Data Variety: The data collected in a smart home should encompass a wide variety of deployed sensors. Typically, this would include an on/off status of a motion sensor, open/closed status of a door sensor, temperature sensor reading, pressure sensor reading, light measurement, etc. The above describes the data variety. Also, the model is designed to be scalable such that data from different sensor technologies can be included to improve prediction or as enhanced sensor technologies are made available in future.

(ii) Big Data Veracity: Sensor data received as a continuous streaming with a variety of attributes should be checked for utmost accuracy. While applying ML approaches, certain attributes such as battery information even though available via streaming, could be ignored initially for training the model. The model is flexible for such attributes to be included based on the requirements to enhance accuracy.

(iii) Big Data Velocity: Since the data collected from each user environment is continuously required to be stored, the number of records collected each day may grow rapidly. Further, as the number of users of a smart home increases, the velocity of the data also increases. The model is developed to provide real-time processing for meeting the data velocity requirements with appropriate big data infrastructure.

(iv) Big Data Value: The value of big data can be realised with our proposed model that brings new data insights and customisation by analysing data patterns of different sensor data pertaining to parameters such as temperature, pressure, humidity, luminescence, movement, time and status. The model provides an enhanced value of Big Data by employing intelligent computations and performing ML of patterns present in sensor datasets to replace manual decisions by automated decisions.

The logical architecture (Figure 2) of our proposed model consists of the first phase of data collection from various home appliances and sensors, as described above. The remaining phases of our model depicted in Figure 2 leverage on effective use of big data technologies. The data preparation phase involves loading the data into a data warehouse for scalable data processing. The next phase is employed for analysing data to detect human behaviour patterns to achieve personalisation. It is used to train the model that learns from a given input data using ML. Further, the final phase involves model evaluation using analytics to result in model prediction for facilitating automated personalised decisions.

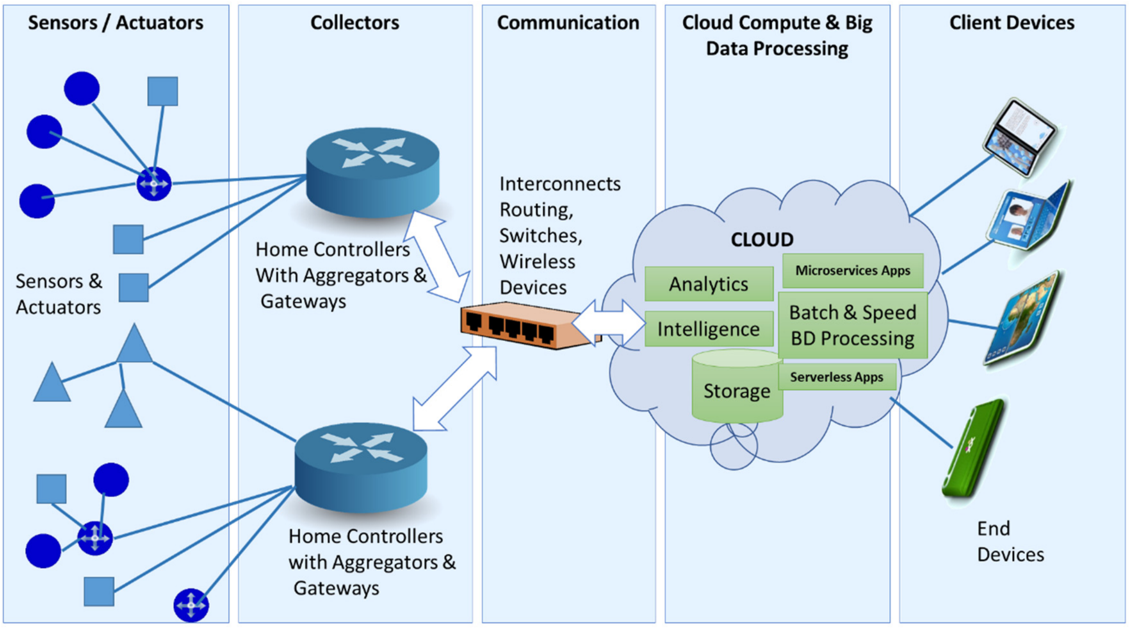

Sensors / Actuators
Collectors
Communication
Cloud Compute & Big Data Processing
Client Devices
Sensors & Actuators
Home Controllers With Aggregators & Gateways
Interconnects Routing, Switches, Wireless Devices
CLOUD
Analytics
Microservices Apps
Intelligence
Batch & Speed BD Processing
Storage
Serverless Apps
Home Controllers with Aggregators & Gateways
End Devices

Figure 1. Generic IoT architecture with cloud and big data processing extensions.

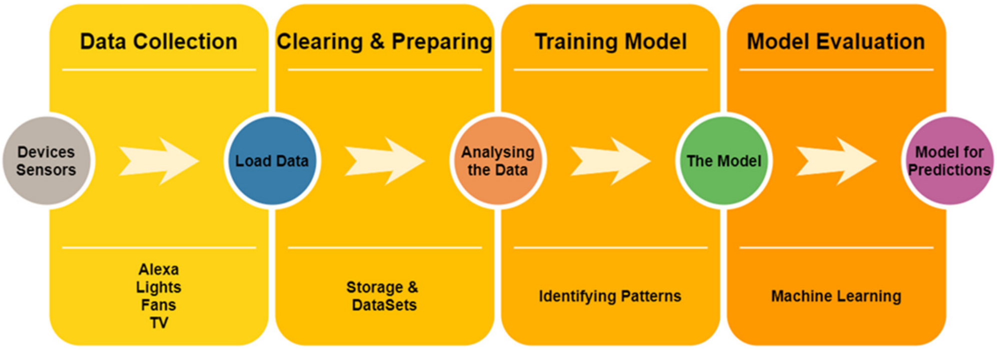


Figure 2. Logical architecture of modelling big data and personalisation.

The data are generated by home devices and sensors by our proposed home automation hub and can be stored into several databases and data stores. The required querying capability is achieved by data warehousing features that are available in cloud software such as Apache Spark QL, Apache Hive or Cloudera Impala. Figure 3 provides a high level physical big data IoT architecture [36–38]. The data analysis and intelligent computations are carried out using distributed cluster computing frameworks such as Apache Sparks. Further, ML algorithms are employed for human behaviour pattern recognition. The result of such intelligent data processing and personalisation can then be archived in the result store and utilised by home devices to carry out automated actions as well as data visualisation. Also, as new data are continuously added via the data stream that gets collected daily, the model makes use of the speed layer and batch layer of big data technologies for data processing. Finally, the output will be used for reinforcement learning in the servicing layer of the big data architecture that caters to both functional and non-functional requirements [39].

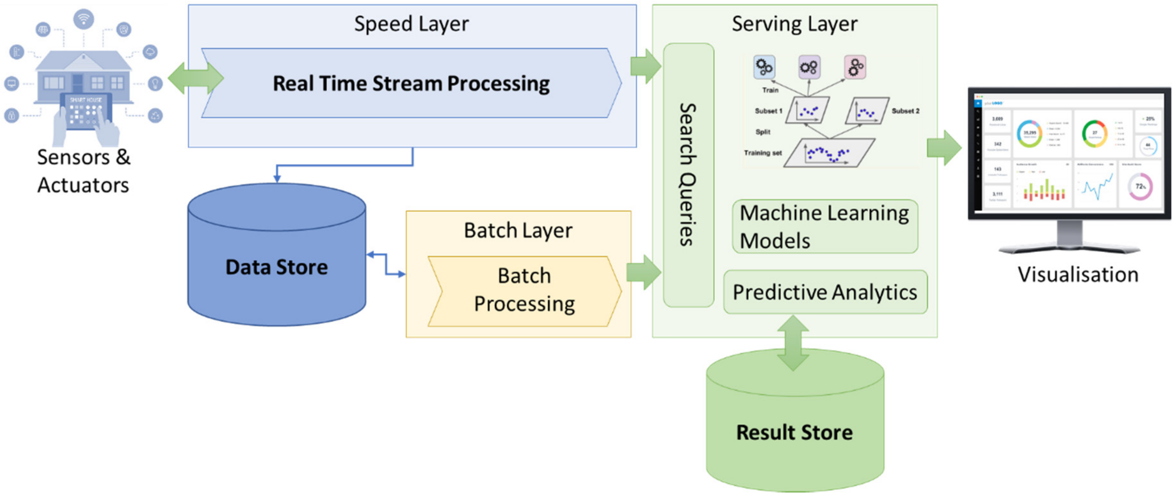


Figure 3. High level physical big data architecture.

4. Use Case of Proposed Model for Smart Home Automation

We consider the use case for the demonstration of our model to detect human activities and behaviour patterns for implementing smart home automation. At its most basic, a smart home is one that uses the so-called "smart" technology to automate and operate important tasks and devices, including lighting, heating/cooling, door locks, doorbells, home security, home entertainment, kitchen appliances and laundry equipment. The smart home deployed IoT devices comprise of sensors such as motion sensors and temperature, humidity and pressure sensors in each room, as well as door sensors. Currently, based on the outcomes of these sensors, it can control other actuators such as ceiling fans, lights, power sockets, air conditioners and TV. Both TV and air conditioners are not directly controlled as they are generally dumb devices but with an IR Blaster, which acts as a thing and can communicate with the intelligent network [40].

A smart technology-based device can sense events around a particular sensor to act autonomously based on this information collected. For example, a movement sensor might sense someone walking into a room and open the shades or turn on the lights or turn up the heat or whatever the actuators are being programmed to do based on the current climate readings. Beyond such home automation tasks, a smart home technology can be applied to learn about the regular habits of the user and use that information to make personalisation efficient and targeted. The "goal" for such devices is to act on mundane operations, thereby freeing the user's valuable time for more important tasks. Smart home technologies not only aspire to automate but also optimise resources where possible. A good example is Apple's Home App available in the commercial market. In addition, smart home hub technologies allow users to control a smart device via a smartphone app. This means users can control smart lighting, heating, and other smart devices even when the user is not at home to use a smartphone for this purpose. Some home hub based controllable devices can also consolidate the control of multiple devices from a variety of manufacturers.

We developed our research prototype as a use case to study the feasibility of the proposed big data processing backed IoT architecture using OpenHab integrator and other smart home devices. Figure 4 shows the physical layout of our smart home research prototype comprising of a living area, a master room, a kitchen, two bedrooms, and a bathroom along with IoT deployment. The proposed architecture supports developing complex cooperative applications that connect, process, store and analyse data received from the edge environment using IoT, cloud and big data technologies [41]. We implemented new applications on scalable platforms containing fully managed services along with ML and big data analytics capabilities for the development of the smart home research prototype.

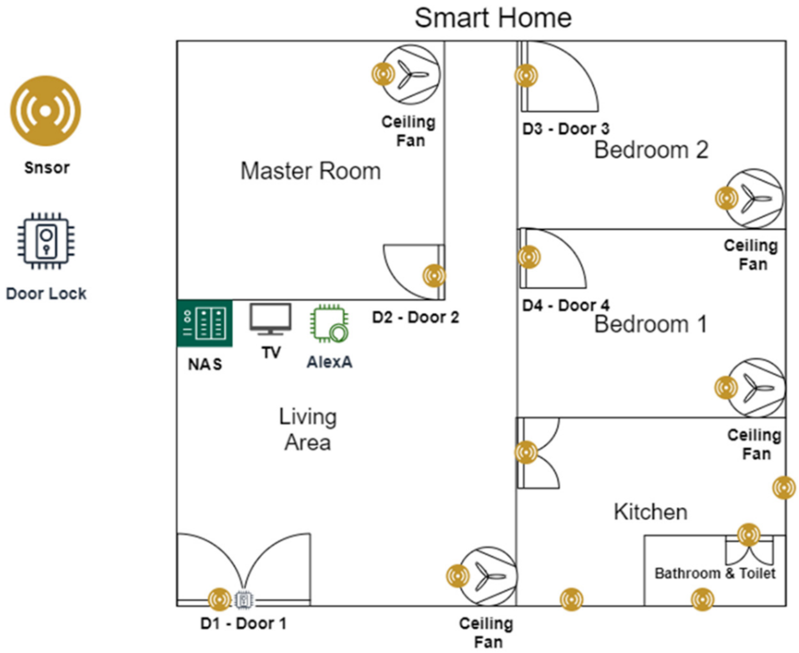


Figure 4. Physical layout of a smart home prototype with IoT deployment.

A unifying architecture needs to be compatible with all available smart devices to overcome challenges, such as, interoperability, abstraction and device heterogeneity. Smart devices comprise of diverse hardware, firmware, and network protocol requirements. Thus, it becomes essential to aggregate these communication patterns via smart hubs. While there are many propriety home controllers such as Apple Homekit, Amazon Alexa and Google

Home, to undergo an unbiased research study, we considered open-source smart hub platforms in providing transparency and protocol interconnect standardisation for achieving a more sustainable and scalable smart home automation Common examples of such smart home hub platforms include single-board-computers such as the Beaglebone, and Raspberry Pi, as well as microcontroller platforms such as the Arduino series, boards from Particle, and the Adafruit Feather. We employed smart hubs to collect information with the aim to convert them into actionable insights that can create richer personalised experiences for smart home residents (users).

We employed Raspberry Pi for developing our smart home research prototype. We adopted Azure IoT Hub, Azure Event Hub and Azure IoT SDK due to its seamless connectivity to Azure streaming and analytics services that can be applied on the collected data. Generally, such smart hubs stay closer to the sensors and actuators communicating via various channels by directly being connected via HTTP based internet protocols. As shown in Figure 5, all these peripheral devices are integrated with an open-source software called OpenHab, which runs on a Linux kernel in Raspberry Pi. As a vendor and technology agnostic platform, OpenHab is an ideal open-source automation software for smart home protype automation meeting our research purpose. The IoT devices generally communicate among themselves in a mesh network whereas the OpenHab smart hub of the network is connected to the main router in a Star topology, thereby forming an overall hybrid topology. With OpenHab Hue bridges, it is possible to construct a plug-and-play architecture of IoT devices. OpenHab supports interconnectivity among the various smart devices. It allows integration via cloud connectors for the most popular cloud-based smart home platforms, including Google Assistant, Amazon Alexa, Apple Home Kit and IFTTT. For our use case, we adopted Samsung, Philips and Hunter based hue bridges for air-conditioning, lights, and fans, respectively; further, we connected these devices to the Azure smart hub that extended auto control of the sensors and actuators.

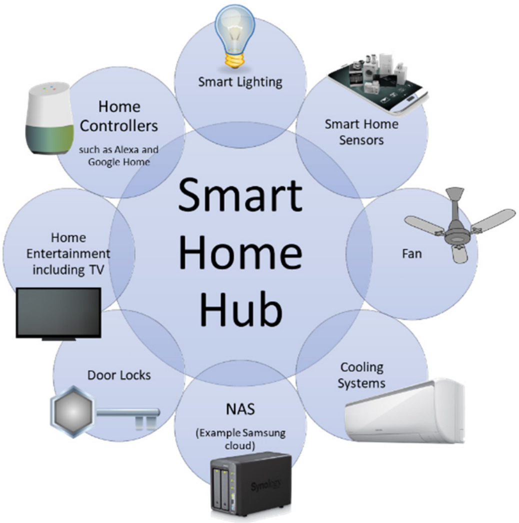
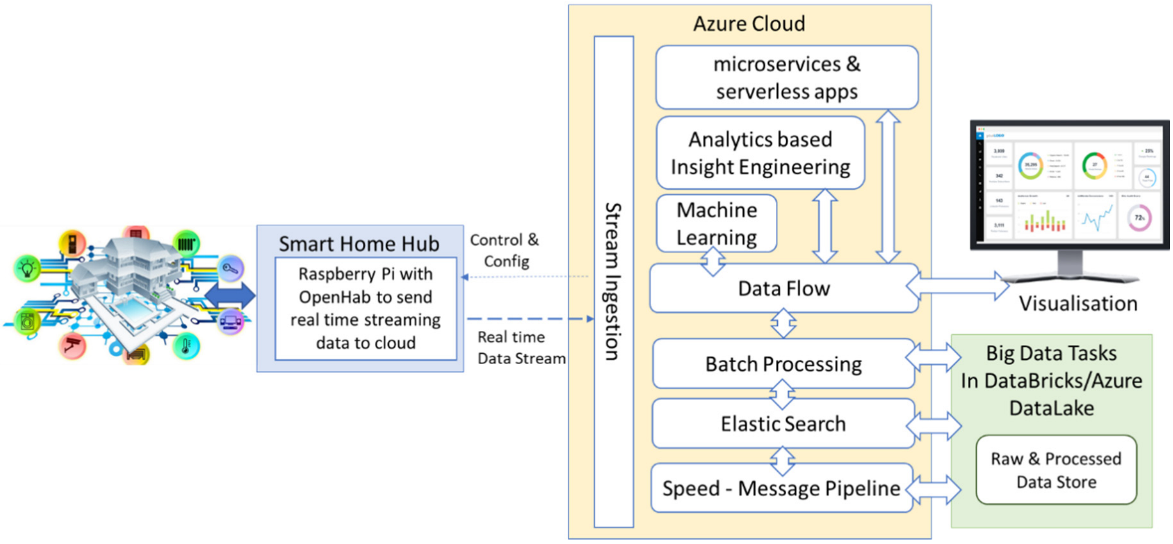


Figure 5. A typical smart home hub in a star topology.

Once we configure the network/device with OpenHab to possess the knowledge about the IoT environment, it is required to constantly keep track of the devices and control them. Using a flexible rule engine and SDK based telemetry, OpenHab communicates to devices with time and event-based triggers, scripts, actions, notifications and voice control. All device tracking data can be logged on a day to day basis. These data logs are further configured to be stored in a persistence data store such as MongoDB, which is hosted in a docker in the network attached storage (NAS). In smart homes, recognition of activity of daily living (ADL) requires creating an artificial intelligence layer overarching the IoT network by leveraging on big data technologies. The challenge here is to achieve accurate automation of unlimited course of actions that a user can perform in a smart home in a non-intrusive manner.

A smart home aims at enhancing the quality of human life covering different domain areas of application such as personal, social, medical, environmental, logistics and energy. With our generic implementation platform, our use case can be extended to cater to different smart devices that include smart plugs and outlets, motion sensors, smart window shades, smart robot-vacuum cleaners, smart cooking appliances, smart doors, home security devices, irrigation system, entertainment devices, heating/cooling devices and fans. Smart devices use various smart networking protocols such as Z-wave, ZigBee, BlueTooth, Insteon, Wi-Fi or older X10. They connect to smart home hub for further inclusive collaboration. This smart hub will connect with multiple compatible smart devices via the home Wi-Fi network; it also connects using Nifi real time streaming service. This real time streaming data are then ingested at the Azure cloud for further processing using DataBricks data lake and spark processing tech stack. Figure 6 shows the key software used in the use case, namely, OpenHab, Azure Cloud Services, Databricks DataLake and Apache Spark Big Data Processing Framework.



Smart Home Hub
Raspberry Pi with OpenHab to send real time streaming data to cloud
Control & Config
Real time Data Stream
Stream Ingestion
Azure Cloud
microservices & serverless apps
Analytics based Insight Engineering
Machine Learning
Data Flow
Batch Processing
Elastic Search
Speed - Message Pipeline
Visualisation
Big Data Tasks In DataBricks/Azure DataLake
Raw & Processed Data Store

Figure 6. Software components employed in smart home hub, cloud and big data processing.

The first step in the use case development involves collecting data from the sensors and using an ML to model a system that can understand user behaviour within the smart home environment. To achieve this, we consider sensors, such as motion sensors, light sensors, door contact sensors, water leak sensors, vibration sensors, fire alarms, smoke sensors, and water leak sensors, that are already deployed and data are being collected. An average of 2900 records of data are stored per day out of which 1600 records can be used for analytics and the rest is used for maintenance purposes.

The collected data consists of the following attributes:

- Item name,
- Attributes like Battery Status, State Type, Last Occurrence, Current Occurrence,
- Value and,
- Time of Occurrence.

In addition to the above, the data collected can be grouped together based on the room in which the event has taken place. So, it would be easier for the analytics engine to understand that all these sensors are located within that room. Moreover, for the purpose of this use case prototype implementation, these sensors have been manually grouped to a room to simplify the process. To demonstrate the application of our model in a real-life setting, we consider a list of sensors (Table 1) that are deployed in the smart home under consideration. The smart home automation will result in the sensors taking decisions for follow-up actions on the IoT devices based on the personalised rules defined for their network, as shown in Table 2.

Table 1. Sensor data and unit of measurement of the use case.

Sensor Data	Unit of Measurement
Humidity	%
Temperature	C
Pressure	KPA
Illumination	Lux
Motion Sensor	On/Off
Motion Sensor Last Motion	Datetime
Door Sensor	OPEN/CLOSED
Door Sensor Last Opened	Datetime
Digital Lock	Lock/Unlock

Table 2. IoT devices and their data.

IoT Devices	IoT Data
TV	Volume/Mute/Power
Switch	On/Off
Lightings	On/Off
Socket	On/Off
Button	Tap
Voice Assistant	Voice Commands
IR Blaster	Commands

A significant value of the IoT deployment is in the interpretation of data and the decisions derived through its analytics. Figure 7 shows the various components of the analytics adopted for implementing the smart home research prototype. The data ingestion component is required for the data to be staged for both stream and batch processing using an event messaging framework. Subsequently, after data gets interpreted and analysed in real time as a streaming dataflow, it is stored in a data lake and retrieved for analytics in the cloud. A stream processing component is required to cater to any continuous stream of data from various devices in real-time, which is possible with in-memory data processing. Batch processing, on the other hand, is efficient in dealing with high-volume data. It is particularly useful when IoT data needs to correlate against historical data. Depending on the use case, our approach allows scalability when the data needs to be correlated with other sources dynamically. In many simple use cases, the data gets logged and dumped to a data lake like Databricks DBFS as the storage component. Subsequently, the prediction and response from data is rendered as outputs where information may be presented in a visual form of a dashboard such as the Databricks dashboard. The system's response can also be sent back to the edge smart devices using the smart hub, where corrective actions can be applied to resolve issues if any.

Data generated by home signals, devices and sensors are gathered in various formats and loaded using technologies such as Apache Flume or Rabbit MQ. An IoT device is usually associated with some sensor or a device in which the purpose is to measure or monitor the physical world. This can be asynchronously performed with respect to the rest of the IoT technology stack. In this use case, we classified three states for human real-time behaviour in a house. At first, we detect user presence in a particular space of the smart home, once detected we progress to identify the activity the user is currently engaged in, and finally, based on the activity we predict the user's next behaviour. A brief explanation for the three-state processing is provided below:

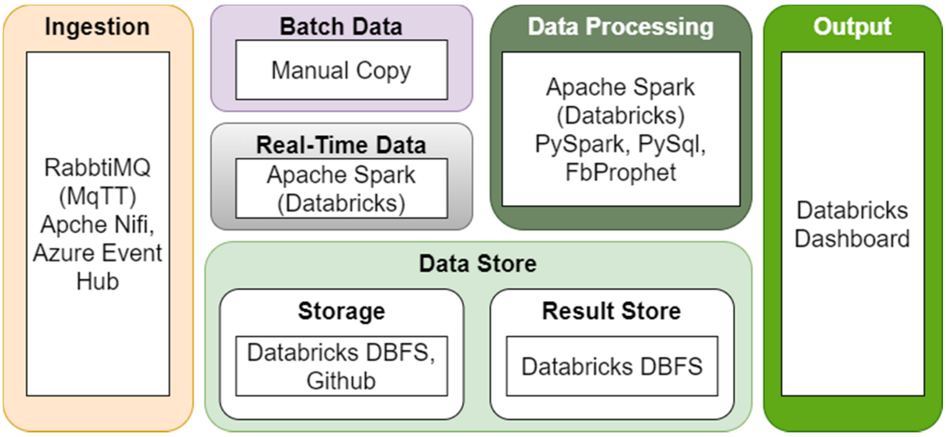


Figure 7. Analytics components using Azure Databricks tools.

4.1. Human Presence Detection

This state classification involves identification of human presence using data such as temperature, humidity and pressure from sensors or data from motion sensors fixed in the selected space or a room. Other information that can be collected regarding the same space of room involves the current state (ON/OFF) of smart devices such as TV, fan, air conditioner and kitchen appliances.

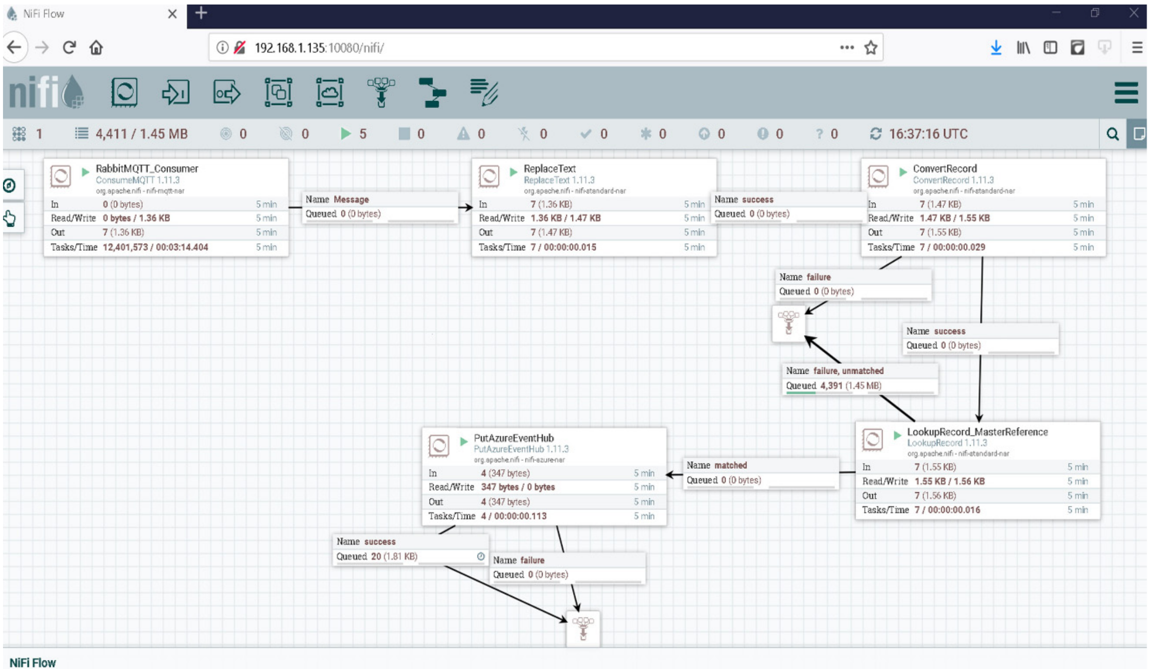
4.2. Human Activity Detection

Human activity in the house such as opening main door, entering a room or space such as living area, kitchen, bathroom, locking or unlocking door, leaving a room or space or the home, etc. Such activities can be detected by interpreting data from devices like door lock, motion sensors in various rooms and selected spaces, door status, main door lock system, etc. Data are collected as event-stream with relevant time-series correlation, and activity detection is performed via pattern recognition of event sequences preset.

4.3. Human Behaviour Detection

Next stage is to predict possible human behaviour based on behaviour analysis of historical data collected. Our analytics model predicts the real time behaviour of user using appropriate ML and analytics algorithms. The predictive model considers day-of-week activity patterns such as weekday and weekend. For instance, a regular Sunday afternoon movie is turned on with a smart entertainment system automatically including other devices such the air conditioner depending on the external weather condition at a particular time when the family members are usually sitting in the living area.

The IoT stream from a sensor to a cloud is assumed to be constantly inputted asynchronously from various devices and are structured or unstructured messages that are processed in real time. Different device controllers emit a combination of structured and unstructured real time stream data continuously that are collected and transmitted by the smart hub into cloud services. For this purpose, we configured Rabbit MQ as a MQTT broker and executed it as a docker container inside the Raspberry Pi. However, transmission to the cloud ingestion component requires a scalable data routing, transformation, and system mediation logic. Thus, the Apache NiFi service (shown in Figure 8) was configured to collect all sensor data via MQTT and transmitted as encrypted content via SSL and HTTPS protocols.



RabbitMQTT_Consumer
ConsumeMQTT 1.11.3
org.apache.nifi - nifi-mqtt-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 1.36 KB	5 min
Out	7 (1.36 KB)	5 min
Tasks/Time	12,401,573 / 00:03:14.404	5 min

Name Message
Queued 0 (0 bytes)

ReplaceText
ReplaceText 1.11.3
org.apache.nifi - nifi-standard-nar

In	7 (1.36 KB)	5 min
Read/Write	1.36 KB / 1.47 KB	5 min
Out	7 (1.47 KB)	5 min
Tasks/Time	7 / 00:00:00.015	5 min

Name success
Queued 0 (0 bytes)

ConvertRecord
ConvertRecord 1.11.3
org.apache.nifi - nifi-standard-nar

In	7 (1.47 KB)	5 min
Read/Write	1.47 KB / 1.55 KB	5 min
Out	7 (1.55 KB)	5 min
Tasks/Time	7 / 00:00:00.029	5 min

Name failure
Queued 0 (0 bytes)

Name success
Queued 0 (0 bytes)

Name failure, unmatched
Queued 4,391 (1.45 MB)

LookupRecord_MasterReference
LookupRecord 1.11.3
org.apache.nifi - nifi-standard-nar

In	7 (1.55 KB)	5 min
Read/Write	1.55 KB / 1.56 KB	5 min
Out	7 (1.56 KB)	5 min
Tasks/Time	7 / 00:00:00.016	5 min

Name matched
Queued 0 (0 bytes)

PutAzureEventHub
PutAzureEventHub 1.11.3
org.apache.nifi - nifi-azure-nar

In	4 (347 bytes)	5 min
Read/Write	347 bytes / 0 bytes	5 min
Out	4 (347 bytes)	5 min
Tasks/Time	4 / 00:00:00.113	5 min

Name success
Queued 20 (1.81 KB)

Name failure
Queued 0 (0 bytes)

Figure 8. Apache NiFi data flow design.

4.4. Ingestion

For ingestion, we adopt a spark streaming framework that supports in-memory operations to avoid the cost of storage in a distributed filesystem. To satisfy IoT real time needs, there is a need to maintain a flow of data and to keep it moving as a pipeline. The data stream will also not be perfect due to missing values (e.g., sensor lost communication), or poorly formed data (e.g., error in transmission), or out of sequence data (data may flow to the cloud from multiple paths). The data ingestion component must scale with event growth and spikes and publish/subscribe API to the interface. Further, approaching near-real-time latency, providing scaling of processing of rules, and supporting data lake storage form essential features for long-term considerations. Data cleansing, filtering, transformation and formatting pipelines are some of the key functionalities to be undertaken before passing the data stream to the processing pipelines.

Azure Event Hubs is a big data streaming platform with an event ingestion service. It can receive and process millions of events per second. Data sent to an event hub can be transformed and stored by using any real-time analytics provider or batching/storage adapters. Event Hub also helps in managing anomaly detection (fraud/outliers), logging, clickstreams, live dashboarding and transaction/telemetry processing. Event Hubs provide a unified streaming platform with time retention buffer, decoupling smart home event producers from big data processing consumers. Thus, for the use case, we employed the Azure Event Hub to implement the event ingestion component for incoming real time data stream. The typical dataset received from the NiFi is shown in Table 3 as an IoT sensor data stream sample.

Table 3. Sample data collected from Apache NiFi.

Sensor Data	Multi-Value List	Timestamp
MasterRoom_TemperatureandHumiditySensor_temperature	31.49, 1582519012	...
MasterRoom_TemperatureandHumiditySensor_humidity	72.73, 1582519012	...
StudyRoom_MotionSensor_motion	0, 1582519041	...
LivingRoom_MotionSensor_illumination	116.0, 1582519051	...
LivingRoom_MotionSensor_motion	1, 1582519051	...
KitchenEntrance_EntranceMotionSensor_motion	1, 1582519070	...
Kitchen_MotionSensor_motion	1, 1582519072	...
GuestRoom_TemperatureandHumiditySensor_temperature	32.12, 1582519119	...
MasterRoom_TemperatureandHumiditySensor_temperature	30.93, 1582519145	...
MasterRoom_TemperatureandHumiditySensor_temperature	31.49, 1582519012	...

4.5. Storage

The collective sensors in the smart home produce continuous data, and a cloud environment ingesting these real time data streams will use a data lake for storage purposes. A data lake provides storage, transaction scope, scalable metadata handling, and unifies streaming and batch data processing upon receiving raw unfiltered data from many sources. Data lakes are flat filesystems, organised hierarchically, with volume, directories, files, and folders. The classic data lake model implemented uses Delta Lake, an open-source storage layer. Delta lake is particularly useful in IoT, as it will store any form of data whether it is structured or unstructured. This bulk persistent mass of data is optimal for data analytics engines. Many analytics algorithms are based on how much data are fed, or how much data are used to train their models.

For simplicity, batch data were stored into GitHub daily and was ingested into Databricks DBFS for prediction modelling in the data analysis component. However, in future implementation, a dedicated database can be used, and scheduler job can fetch the data for prediction. Currently, collected data are further pre-processed, filtered, enriched, segregated, and transferred into meaningful structured sets of data that are relevant for finding out patterns for human real-time behaviour in a house.

4.6. Analytics Processing

At the initial stage of our prototype development, we employed batch processing to find patterns and identify features for ML. However, for the advanced behaviour predictions, more sophisticated ML and prediction models were necessary. As far as real-time or near real-time data analysis of human behaviour at a home/room is concerned, steam processing is a more convenient way of processing data whereby live streaming data can be fed to the training model, so that the system can learn to process patterns and evolves to be better with time.

For the prototype development, we adjust smart home parameters such as temperature and lighting using regressors. The model intends to forecast such patterns during real time to make the necessary adjustments in the parameters. Our prototype is developed based on few weeks of data collected in a real-world setting. While ARIMA models can include seasonal covariates, adding these covariates required more expertise on various temperature patterns. Hence, with the aim of applying ML and achieving value through analytics, we used the FBProphet model to meet our research objectives. FBProphet is a forecasting time series additive model where non-linear trends are fit with weekly seasonality plus holiday effects. FBProphet is robust and caters to missing data, trend-shift, and any other outlier very well. In this research study, we decomposed the time series model with three main components: trend, weekly seasonality, and holidays. They are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

where $g(t)$ is the trend function, which models non-periodic changes in the value of the time series, $s(t)$ represents periodic changes (e.g., here it is weekly seasonality), and $h(t)$ represents the effects of holidays which occur on potentially irregular schedules over one or more days. The error term ϵ_t represents any idiosyncratic changes that are not accommodated by the model. The parametric assumption is that ϵ_t is normally distributed. Our model is a generalised additive model, a class of regression models with potentially non-linear smoothers applied to the time regressor. The measurements do not need to be regularly spaced, and we do not need to interpolate missing values or removing outliers. Thus, the model best suits our prototype needs, and we could accommodate seasonality with multiple periods to understand the trends. Data were collected for three months and the patterns were analysed on a weekly (seasonal) basis from March to May of 2020. Even though the data sensing, actuating changes and final prediction vs. actual comparison are done at different intervals, they form typical data requirements for the use case development. This initial research prototype design does not include aggressors, which would be considered in future work as part of our ongoing research. A sample prediction result obtained is shown in Table 4.

Table 4. Sample forecasting results using FBProphet.

Item	Timestamp	Value	Type
StudyRoom_Temperature	2020-04-16 02:45	28.24911524405057	Forecast
StudyRoom_Temperature	2020-04-16 02:46	28.2490169862983	Forecast
StudyRoom_Temperature	2020-04-16 02:47	28.24891872854603	Forecast
StudyRoom_Temperature	2020-04-16 02:48	28.248820470793763	Forecast

From the initial analysis we categorise and understand data patterns by performing pattern recognition, rule engine formulations and activity insights. Based on these preliminary event patterns, time sequence of events and device states, we arrived at the feature engineering aspects of an activity detection and behaviour changes model. We further used these features and ML for prediction of behaviour. Databricks is an environment that makes it easy to build, train, manage, and deploy deep learning models at scale. Databricks integrates tightly with popular Spark open-source libraries and with the ML flow machine learning platform API to support the end-to-end ML lifecycle from data preparation to deployment. Repeat the various activity event stream ingestion to train the model by adding features to let the model learn from new data if any. We considered the implementation platforms based on the performance evaluation reported in literature [42,43]. Initially data are split into trainings and testing subsets for applying ML algorithms to identify patterns to drive a model. By running a set of algorithms/feature-sets that are relevant for detecting human behaviours at home to train a model that leans a set of algorithms/parameters from the home's data. Further, the model is validated for its performance and to check the desired output for a variety of relevant input datasets and evaluated. Thereby, the model can then be deployed in a live environment for predications once the model is validated to perform as expected. Our initial research prototype has a focus more on the real time ML implementation and is scalable for expansion to study the impact of deep learning as future work.

The data pipeline processing codes were retrieved from GitHub code repository as stored notebooks. The interactions of smart home and Azure Cloud along with Databricks Delta Lake are shown in Figure 9. The Apache Spark framework was employed to process big data stored on Databricks DBFS, and the cluster setup included services such as memory pool, scheduler, cluster manager, job tracker, DAG tracer, memory and workload monitoring. Databricks Runtime ML was implemented and updated with every Databricks Runtime release.

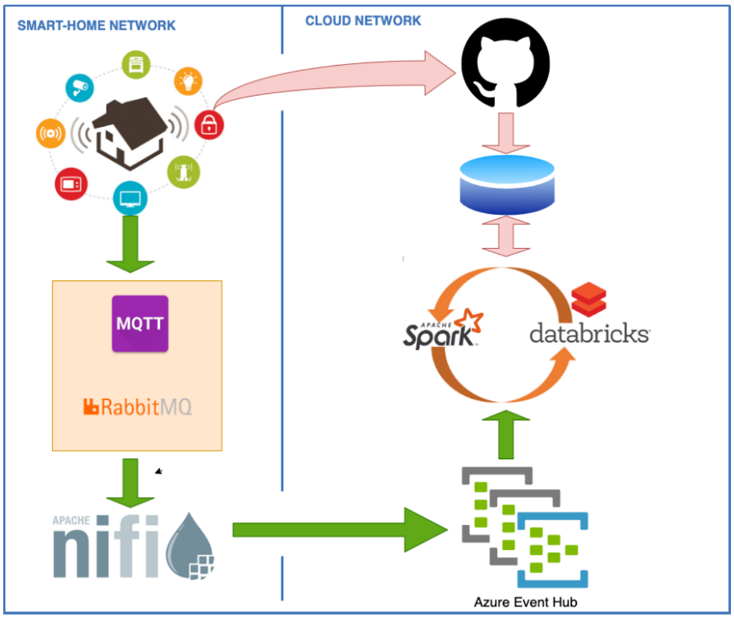


Figure 9. Interactions among a smart home and Azure Databricks.

Spark Streaming is a distributed data stream framework for processing live data streams in real time. Spark stream frames enabled the processing of high-velocity data stream by allowing a combination of current data streams with historical data stream for pattern recognition. Frames derived from past data analysis for detecting any human activities were executed for pattern matching against a live data stream to determine any real time human activity in the entire home. Similarly, activity detection in the entire home in real time, such as detecting human presence in a room or a space was achieved using queries on live stream event patterns. A sample streaming comparison table on the temperature parameter with our prototype deployment is shown in Table 5.

Table 5. Sample streaming comparison for the temperature parameter.

Item	Timestamp	Forecast	Value
StudyRoom_TemperatureandHumiditySensor_temperature	2020-04-16T19:03	28.15301916233153	29.62
StudyRoom_TemperatureandHumiditySensor_temperature	2020-04-16T19:03	28.15301916233153	29.76
StudyRoom_TemperatureandHumiditySensor_temperature	2020-04-16T19:04	28.152920904579258	29.74
StudyRoom_TemperatureandHumiditySensor_temperature	2020-04-16T19:04	28.152920904579258	29.67

While Table 4 shows the prediction model derived using FbProphet, Table 5 provides a comparison of the prediction with the actual data from the smart home setup using the event stream that was filtered for a particular room. In addition, Table 5 observes the effectiveness of actuators. Usage of this comparison in Table 5 is to identify noticeable difference between the two values (forecast vs. actual). This comparison is used to communicate back to the Home Automation system within user premises to initialise the corrective measures. For example, if a predicted value is 26 °C but the actual value is 28 °C, the insight derived from the comparison calls for an action that the monitored zone needs to be cooled down by 2 °C, thereby activating the air conditioner via another message bus. Similarly, other attributes like illumination, pressure, humidity were predicted and evaluated in other zones or rooms.

Identifying and realising activity and device usage patterns were challenging tasks. The following batch processing pipelines were particularly useful in building a pattern catalogue that was later used to train the ML model:

(1) Activity patterns based on past three months of human activity and device usage data sets.
(2) Human presence patterns. We used spark query to detect a data frame that indicates human presence in a room, and the same data frame was matched by the streaming algorithm to detect real-time human presence in a room or space.
(3) Temperature pattern. As a trial, a simplified time series prediction model was generated for temperature in a selected space (Study Room) using FBProphet without adding any aggressors, which are dependencies to a model that can affect the predictions, and then stored in a Databricks table. Similarly, other attributes such as illumination, pressure, and humidity patterns were created to be used during prediction.

Streaming systems are "lighter" in their semantics expressed as low-latency analytics. Spark streaming APIs implement analytics using algorithms that are limited in heuristics. In the use case under consideration, we adopted a known competitive ratio for the streaming algorithm, and the resulting smart home device adjustment performance was acceptable. In industry scale projects, more sophisticated in-memory data storage and low latency stream processing pipelines are recommended.

5. Discussion

As mentioned earlier, creating an ML model is usually a batch-process that uses historical data to train a statistical model. As soon as that model is available, it can be used to score new streaming data to obtain an estimate of the specific aspect for which the model was trained. The unified structured APIs of Apache Spark across batch, ML, and streaming make it straightforward to apply a previously trained model to a streaming DataFrame. For instance, to determine the human presence in a particular space or room, we use a combination of factors such as movement sensing, temperature, humidity, lighting, and carbon emission. The occupancy information in the training dataset was obtained using camera images of the room to detect with certainty the presence of people in it. Using these data, we trained a logistic regression model that estimated the occupancy, represented by the binomial outcome [0, 1], where 0 = not occupied and 1 = occupied.

Another counter validation was performed using time interval between event occurrences such as thermostat getting triggered or lighting state changed, or movement change discovered. All these were incorporated into the Spark streaming pipeline as shown in Figure 10. In the ML process described earlier, we made a distinction between the learning and scoring phases, in which the learning step was mainly an offline process.

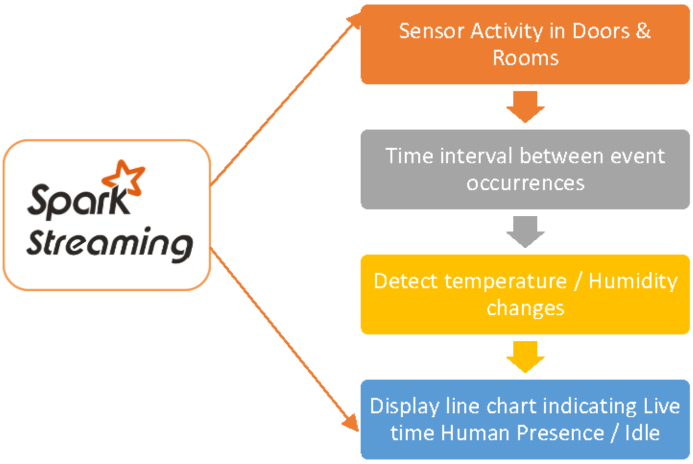

Spark Streaming
Sensor Activity in Doors & Rooms
Time interval between event occurrences
Detect temperature / Humidity changes
Display line chart indicating Live time Human Presence / Idle

Figure 10. Spark Stream processing pipeline for activity patterns.

Based on a human presence in a certain location or room, the home appliances will auto respond. An example of the response actions signals communicated to actuators using our prototype is presented in Table 6. ML applications need to use shared storage for data loading and model checkpointing. We loaded tabular ML data from CSV files. We converted DataFrames into pandas for regression calculations and to NumPy format for visualisation purposes. Using non-intrusive sensors, various data insights are obtained, and statistical information is analysed. Examples include monthly home activity summaries, human presence reports for selected intervals, various room occupancy on different time slots, and based on weather parameters including temperature, humidity, illumination, atmospheric pressure, etc. The hue bridge and Wi-Fi based smart home controller used can be activated via voice. The actuators used in the prototype were also controlled by the recorded voice commands played by the bridge to switch the lights and fan systems on/off, and also to set the air-conditioner to a designated temperature. In Table 6, the column 'Item' refers to the sensor under observation. The column 'actuator signal' refers to the voice enable command such as ON/OFF/SET TEMP. Another column 'is present' records human presence and the column 'is idle' tracks human activity, if any.

Table 6. Sample action signals to actuators.

Item	Timestamp	Actuator Signal	Is Present	Is Idle
StudyRoom_MotionSensor_motion	...	1.0	1	0
StudyRoom_DoorSensor_isOpen	...	0	0	0
LivingRoom_TemperatureandHumiditySensor_temperature	...	30.19	0	1
LivingRoom_TemperatureandHumiditySensor_pressure	...	100.08	0	1

It was also important to decide on the implementation factors for the data pipeline layers such as ingestion, collection, wrangling, messages queues, analysis, storage, and access to new data insights. We performed code execution in Jupyter notebook using PySpark and integrated data frame, ML, statistics and other related packages that come with Spark (Databricks) to perform data analytics. Based on the data originating from IoT devices and sensors and the data stream collection, we employed appropriate storage strategy. We adopted the big data architecture for designing the data ingestion layer, data wrangling layer and processing layer based on the networking protocols and storage requirements.

Among the several challenges considered in this prototype development was the selection of the most suitable model for time series analysis. We explored different time series modelling and how it can be improved to suit our data considerations. Initially, we adopted for time series analysis, but this required data to be continuous in nature with regular intervals, otherwise, it had to be interpolated. However, this was resolved by experimenting and employing FBProphet for achieving the best-fit time series analysis as it does not require the time series to be a continuous function.

There were limitations in the big data platform implementation, as we used the Databricks Community Edition, clusters timed out after six hours of usage and new clusters had to be created. This required all the software library components on the cluster to be installed again. It also posed limitation in querying streaming data and the restriction is to use queries with joins performed on static data. Further, the event bus in Databricks faced frequent disconnect and had to be resolved by implementing different consumer groups for each individual stream.

This work is the first step in developing a prototype in modelling big data and personalisation in a novel non-intrusive approach to achieve smart home automation. Future work will involve improving the model learning aspect by providing dedicated jobs and data storage for ML based on past data as well as from predicted data store for analysis and feedback. It would also involve adding more dependencies and tuning ML modelling and benchmarks based on improvements in accuracy and real-time performance. It would also be useful to incorporate feedback to the user's home automation system through a message bus and to use this information for enhancing further actions/decisions.

6. Conclusions and Future Research Directions

'Behind every cloud is another cloud', Judy (circa 1922-69) quoted satirically on life. Real time analytics via big data, cloud and its connectivity to things can be considered as today's wizardry of Oz. The proliferation of smart devices with collaborative and communicating capabilities make smart sensing blend seamlessly into actionable insights. Internet of Things is popular for interconnecting things such as devices, sensors, equipment, software, and information services. On the other hand, big data processing enables deeper analytics from a heterogenous pool of data streamed continuously. The novel integration of IoT along with big data analytics using open standards has created scope for innovative opportunities in a smart home context. Real time analytics when implemented independently enables the on-demand smart solutions with the least latency. On the other hand, IoT helps with edge device connectivity and keeping everything lightweight all the way. Together, they complement each other, making synergy towards building an even smarter home.

This paper presented a unique modelling approach for technology instantiation of big data, IoT and non-intrusive personalisation by using ML. Further, we demonstrated its practical implementation in a smart home environment as a use case. First, we proposed an architecture for IoT based big data analytics. Next, we explored a typical smart home model to extend its capabilities by tapping into cloud and big data analytics solutions. The relationship between real time systems and IoT was also discussed. In addition, we presented the relevant ML processing model for real time data stream, implementation, framework, and technologies. A smart home implementation was also provided as a use case of our proposed modelling of big data and personalisation using ML.

Overall, our novel approach to collect and integrate non-intrusive data from open standard IoT devices was successfully implemented to achieve a smart home automation by employing big data analytics and ML of user presence, user activity and user behaviour. By integrating technologies with a non-intrusive approach, smart home automation could provide a personalised assisted living but could be confined rigidly within a specific home environment. Future research will focus on extending our technology instantiation model to incorporate deep learning and reinforcement via layers of Spark DAGs. In addition to the current components such as trend, seasonality, and holidays, our future research study would explore the effects of using aggressors. Future work will also focus on evaluating the flexibility and adaptability of the proposed model to different home environments. The big picture of progressing the model from smart homes to smart communities will be explored.

Author Contributions: Conceptualisation, S.P.R.A. and R.V.; methodology, S.V.; investigation, S.P.R.A.; validation, R.V.; writing—original draft preparation, S.P.R.A., R.V. and S.V.; writing—review and editing, S.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval are not required for this study due to non-interventional data collected automatically by sensors, and the authors who themselves participated as subjects in this prototype study have given their consent to publish. In addition, local anonymization was applied on all data prior to public cloud processing to prevent unauthorized access or identification of any information that may be considered sensitive.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data available on request from the authors.

Acknowledgments: This work contains processed information from static and dynamic datasets from a smart home setting with smart hub running on a docker container with Raspberry Pi, OpenHab and Apache NiFi. The authors thank the students (Aboobacker M.S.S.M Hafeez, Vimalarajah Thavarajah and Kunal Jadhav) for taking part in the research study that helped in the use case implementation. The authors also wish to acknowledge the reviewers and editor for their invaluable inputs and support rendered for this paper.

Conflicts of Interest: The authors declare no conflict of interests.

References

1. Helal, S.; Bull, C.N. From Smart Homes to Smart-Ready Homes and Communities. *Dement. Geriatr. Cogn. Disord.* **2019**, *47*, 157–163. [CrossRef] [PubMed]
2. Stojkoska, B.L.R.; Trivodaliev, K.V. A review of Internet of Things for smart home: Challenges and solutions. *J. Clean. Prod.* **2017**, *140*, 1454–1464. [CrossRef]
3. Hui, T.K.; Sherratt, R.S.; Sánchez, D.D. Major requirements for building smart homes in smart cities based on internet of things technologies. *Futur. Gener. Comput. Syst.* **2017**, *76*, 358–369. [CrossRef]
4. Benmansour, A.; Bouchachia, A.; Feham, M. Multioccupant Activity Recognition in Pervasive Smart Home Environments. *ACM Comput. Surv.* **2016**, *48*, 1–36. [CrossRef]
5. Vardakas, J.S.; Zorba, N.; Verikoukis, C.V. Scheduling policies for two-state smart-home appliances in dynamic electricity pricing environments. *Energy* **2014**, *69*, 455–469. [CrossRef]
6. Choi, D.; Choi, H.; Shon, D. Future changes to smart home based on AAL healthcare service. *J. Asian Arch. Build. Eng.* **2019**, *18*, 190–199. [CrossRef]
7. Russell, L.; Goubran, R.; Kwamena, F. Personalization Using Sensors for Preliminary Human Detection in an IoT Environment. In Proceedings of the 2015 International Conference on Distributed Computing in Sensor Systems, Fortaleza, Brazil, 10–12 June 2015; pp. 236–241.
8. Desai, D.; Hardik, U. Security and Privacy Consideration for Internet of Things in Smart Home Environments. *Int. J. Eng. Res. Dev.* **2014**, *6*, 73–83.
9. Li, R.Y.M.; Li, H.C.Y.; Mak, C.K.; Tang, T.B. Sustainable Smart Home and Home Automation: Big Data Analytics Approach. *Int. J. Smart Home* **2016**, *10*, 177–198. [CrossRef]
10. Overmars, A.; Venkatraman, S. Towards a Secure and Scalable IoT Infrastructure: A Pilot Deployment for a Smart Water Monitoring System. *Technologies* **2020**, *8*, 50. [CrossRef]
11. Lei, X.; Tu, G.; Liu, A.X.; Li, C.; Xie, T. The Insecurity of Home Digital Voice Assistants-Vulnerabilities, Attacks and Countermeasures. In Proceedings of the 2018 IEEE Conference on Communications and Network Security (CNS), Beijing, China, 30 May–1 June 2018; pp. 1–9.
12. Fortin-Simard, D.; Bilodeau, J.-S.; Bouchard, K.; Gaboury, S.; Bouchard, B.; Bouzouane, A. Exploiting Passive RFID Technology for Activity Recognition in Smart Homes. *IEEE Intell. Syst.* **2015**, *30*, 7–15. [CrossRef]
13. Ding, D.; Cooper, R.A.; Pasquina, P.F.; Fici-Pasquina, L. Sensor technology for smart homes. *Maturitas* **2011**, *69*, 131–136. [CrossRef] [PubMed]
14. Xu, L.; Pombo, N. Human Behavior Prediction Though Noninvasive and Privacy-Preserving Internet of Things (IoT) Assisted Monitoring. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (Wf-Iot), Limerick, Ireland, 15–18 April 2019; pp. 773–777.
15. Zhang, Y.; Yang, G.; Ma, S. Non-intrusive Load Monitoring based on Convolutional Neural Network with Differential Input. *Procedia CIRP* **2019**, *83*, 670–674. [CrossRef]
16. Asri, H.; Mousannif, H.; Al Moatassime, H. Reality mining and predictive analytics for building smart applications. *J. Big Data* **2019**, *6*, 66. [CrossRef]
17. Hassani, H.; Huang, X.; Silva, E.S. Big Data and Climate Change. *Big Data Cogn. Comput.* **2019**, *3*, 12. [CrossRef]
18. Hamid, O.; Barbarosou, M.; Papageorgas, P.; Prekas, K.; Salame, C.-T. Automatic recognition of electric loads analyzing the characteristic parameters of the consumed electric power through a Non-Intrusive Monitoring methodology. *Energy Procedia* **2017**, *119*, 742–751. [CrossRef]
19. Pattar, S.; Kulkarni, D.S.; Vala, D.; Buyya, R.; Venugopal, K.R.; Iyengar, S.; Patnaik, L. Progressive Search Algorithm for Service Discovery in an IoT Ecosystem. In Proceedings of the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 14–17 July 2019; pp. 1041–1048.
20. de Brouwer, M.; Ongenae, F.; Bonte, P.; de Turck, F. Towards a Cascading Reasoning Framework to Support Responsive Am-bient-Intelligent Healthcare Interventions. *Sensors* **2018**, *18*, 3514. [CrossRef]
21. Ismail, A.; Abdlerazek, S.; El-Henawy, I.M. Development of Smart Healthcare System Based on Speech Recognition Using Support Vector Machine and Dynamic Time Warping. *Sustainability* **2020**, *12*, 2403. [CrossRef]
22. Elsokah, M.M.; Saleh, H.H.; Ze, A.R. Next Generation Home Automation System Based on Voice Recognition. In Proceedings of the 6th International Conference on Engineering & MIS 2020, Almaty, Kazakhstan, 14–16 September 2020; pp. 1–7.

23. Jia, M.; Srinivasan, R.S.; Raheem, A.A. From occupancy to occupant behavior: An analytical survey of data acquisition technologies, modeling methodologies and simulation coupling mechanisms for building energy efficiency. *Renew. Sust. Energ. Rev.* **2017**, *68*, 525–540. [CrossRef]
24. Kim, K.; Jalal, A.; Mahmood, M. Vision-Based Human Activity Recognition System Using Depth Silhouettes: A Smart Home System for Monitoring the Residents. *J. Electr. Eng. Technol.* **2019**, *14*, 2567–2573. [CrossRef]
25. Penha, D.D.P.; Castro, A.R.G. Home Appliance Identification for Nilm Systems Based on Deep Neural Networks. *Int. J. Artif. Intell. Appl.* **2018**, *9*, 69–80. [CrossRef]
26. Wilkowska, W.; Ziefle, M.; Himmel, S. *Perceptions of Personal Privacy in Smart Home Technologies: Do User Assessments Vary Depending on the Research Method?* Springer: Berlin/Heidelberg, Germany, 2015; Volume 9190, pp. 592–603.
27. Daissaoui, A.; Boulmakoul, A.; Karim, L.; Lbath, A. IoT and Big Data Analytics for Smart Buildings: A Survey. *Procedia Comput. Sci.* **2020**, *170*, 161–168. [CrossRef]
28. Fabi, V.; Spigliantini, G.; Corgnati, S.P. Insights on smart home concept and occupants' interaction with building con-trols. *Energy Procedia* **2017**, *111*, 759–769. [CrossRef]
29. Vallée, T.; Sedki, K.; Despres, S.; Jaulant, M.-C.; Tabia, K.; Ugon, A. On personalization in IoT. In Proceedings of the 2016 International Conference on Computational Science and Computational Intelligence, Las Vegas, NV, USA, 15–17 December 2016.
30. Zhao, A.; Qi, L.; Li, J.; Dong, J.; Yu, H. A hybrid spatio-temporal model for detection and severity rating of Parkinson's disease from gait data. *Neurocomputing* **2018**, *315*, 1–8. [CrossRef]
31. Garcia, F.C.C.; Creayla, C.M.C.; Macabebe, E.Q.B. Development of an intelligent system for smart home energy dis-aggregation using stacked denoising autoencoders. *Procedia Comput. Sci.* **2017**, *105*, 248–255. [CrossRef]
32. Sanchez, L.; Lanza, J.; Santana, J.R.; Agarwal, R.; Raverdy, P.G.; Elsaleh, T.; Fathy, Y.; Jeong, S.; Dadoukis, A.; Korakis, T.; et al. Federation of Internet of Things Testbeds for the Realization of a Semantically-Enabled Multi-Domain Data Marketplace. *Sensors* **2018**, *18*, 3375. [CrossRef]
33. Ta, V.-D.; Liu, C.-M.; Nkabinde, G.W. Big data stream computing in healthcare real-time analytics. In Proceedings of the 2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, 5–7 July 2016; pp. 37–42.
34. Atote, B.S.; Zahoor, S.; Dangra, B.; Bedekar, M. Personalization in user profiling: Privacy and security issues. In Proceedings of the 2016 International Conference on Internet of Things and Applications (IOTA), Pune, India, 22–24 January 2016; pp. 415–417.
35. Chen, R.; Tong, Y.; Yang, J.; Wu, M. Residual Reconstruction Algorithm Based on Sub-pixel Multi-hypothesis Prediction for Distributed Compressive Video Sensing. In Proceedings of the 12th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS-2018), Matsue, Japan, 4–6 July 2018; pp. 599–605.
36. Singh, K.N.; Behera, R.K.; Mantri, J.K. Big Data Ecosystem: Review on Architectural Evolution. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 335–345.
37. Persico, V.; Pescapé, A.; Picariello, A.; Sperli, G. Benchmarking big data architectures for social networks data processing using public cloud platforms. *Futur. Gener. Comput. Syst.* **2018**, *89*, 98–109. [CrossRef]
38. Lin, J. The Lambda and the Kappa. *IEEE Internet Comput.* **2017**, *21*, 60–66. [CrossRef]
39. Vergilio, T.; Ramachandran, M. Non-functional Requirements for Real World Big Data Systems—An Investigation of Big Data Architectures at Facebook, Twitter and Netflix. In Proceedings of the Proceedings of the 13th International Conference on Software Technologies, Porto, Portugal, 26–28 July 2018; pp. 833–840.
40. Dinesh, C.; Makonin, S.; Baji´c, I.V. Residential power forecasting using load identification and graph spectral 25 clustering. *IEEE Trans. Circuits Syst. II Express Briefs* **2019**, *6*, 1900–1904. [CrossRef]
41. Rassam, L.; Zellou, A.; Rachad, T. *Towards a Big Data Architecture for Heterogeneous Data Sources*; EasyChair: Manchester, UK, 2020.
42. Singh, A.; Khamparia, A.; Luhach, A.K. Performance comparison of Apache Hadoop and Apache Spark. In Proceedings of the Third International Conference on Advanced Informatics for Computing Research—ICAICR '19, Shimla, India, 15–16 June 2019; pp. 1–5.
43. Oliveira, F.R.; Cura, L.M.D.V. Performance Evaluation of NoSQL Multi-Model Data Stores in Polyglot Persistence Applications. In Proceedings of the 20th International Database Engineering & Applications Symposium on—IDEAS '16, Montreal, QC, Canada, 11–13 July 2016; pp. 230–235.