American Journal of Applied Sciences

Original Research Paper

Real Time Object Tracking and Protection: Hybrid Logistics Architecture in Ubiquitous Environments (HLAiUE)

[1]Abdul Jawwad Patel, [1]Abeer Alsadoon, [1]P.W.C. Prasad and [2]A. Elchouemi

[1]School of Computing and Mathematics, Charles Sturt University, Sydney, Australia
[2]Walden University, USA

Article history
Received: 01-05-2018
Revised: 03-09-2018
Accepted: 16-10-2018

Corresponding Author:
P.W.C. Prasad
School of Computing and
Mathematics, Charles Sturt
University, Sydney, Australia
Email: cwithana@studygroup.com

Abstract: Realtime object security and vehicle tracking have not been successfully implemented in commercial vehicles due to limitations in processing large amount of data and hardware capabilities. This paper aimed to improve the object security, processing power and data storage in a distributed environment. The proposed system (HLAiUE) consists of a three-layer architecture, first layer is used to collect data, second layer distributes the data processing and the last layer stores the processed data. The designed architecture improves the performance and security by applying data compression algorithms in a real time and removes the hardware dependency by processing the data in a distributed environment using Hadoop and Spark framework. The results demonstrate that the proposed architecture improves processing time by 80% compared to other algorithms and provides more reliability, security and flexibility due to ubiquity and the absence of hardware dependency in comparison with other existing architectures which are hardware dependent. Also, using an NOSQL database server in a distributed environment optimizes data storage by up to 80% and is flexible because all the infrastructure is in a distributed environment. The proposed architecture improves the object tracking by implementing compression technique in a distributed architecture. Thus, this system improves the accuracy up to 80% without depending on hardware resources.

Keywords: Object Detection, Correlation Coefficient, Occlusion, Augmented Reality, Construction Site

Introduction

Nowadays it is essential for almost every organization to use some forms of technology (Valentine and Stewart, 2013; Cao *et al*., 2014) to increase efficiency, productivity and provide the best value to the customer to survive and stay competitive in the market (Leten *et al*., 2016). With increased popularity and availability of the Internet, many traditional logistics companies have improved their business by enhancing their technological capabilities through replacing traditional with electronic commerce (e-commerce) (Maity and Dass, 2014) or mobile commerce (m-commerce) (Yoshii and Sumita, 2016) and are now moving towards ubiquitous computing (u-computing) (Chung, 2014). Ubiquitous computing provides an environment where many actors such as software agents, customers, devices, etc. interact with each other anytime, anywhere because of network ubiquity, universality, uniqueness and unison (Weiser, 1993).

The goal of utilizing technologies in business is to maximize the satisfaction level of end users. Logistics plays a major role in this because every organization is involved either in Business to Business (B2B) (Vlachos *et al*., 2016), Business to Consumer (B2C), Consumer to Consumer (C2C) (Paris *et al*., 2016) or Government to Government (G2G) (Christou and Michalakos, 2010) requires logistics services for the movement of goods, people or services from one place to another and the value of these resources can be measured in millions of dollars (Min and Joo, 2009). Due to the high value of products, it has become essential for logistic companies to utilize every possible technology to secure these goods while they are in transit (Dawe, 1994).

There are several devices and Internet Of Things (IOT) based applications that are required to make transportation secure and reliable (Bandyopadhyay and Sen, 2011). These include radio frequency, the Internet, global positioning systems, wireless radios, satellites, databases, network devices, a cellular network and software applications, etc. Logistics and supply chain based companies collect vast amounts of data gathered from a variety of sources installed in a vehicle and the sole purpose of these data is to provide real-time information to the user, management and a variety of other actors involved directly or indirectly in business transactions (Giannakis *et al*., 2016). The collected data first need to be processed by integrating business scenarios so that they can be converted into information Due to large size and complex nature of data, Imani and Braga-Neto (2018) proposed approximate MMSE filtering and smoothing algorithms to manage computational and memory requirement to enhance performance, where as, several other authors (Gong *et al*., 20018; Jin *et al*., 2018; Ghoreishi and Allaire, 2017) proposed optimization algorithms which can be used to reduce cost increase efficiency and reliability (Su *et al*., 2014).

Data processing for structured and unstructured data is the key element for tracking vehicle and ensuring vehicle positioning and reporting back to the system. In order to get optimal results, several factors need to be considered including the amount and the size of data each device or server will process and how to utilize the available resources in their full capacity. The key contributing factor includes the devices used to get the data, how the collected data can be processed, after processing the data what information can be extracted and how the system can report back to user based on the available information.

In this study, we will first discuss the fundamental challenges logistics companies are facing regarding data processing, storing of large amounts of unstructured data and basic building blocks of security including physical as well as data security. Furthermore, we will discuss proposed architectures to solve these issues and their limitations.

Related Work

Studies (Musa *et al*., 2014; Li and Xiao, 2013; Hogpracha and Vongpradhip, 2015; Qi *et al*., 2015) have identified several key problems with different logistics architectures in which the most critical are discussed below.

Issues with Logistics Architecture in Ubiquitous Environments

Object positioning and tracking is one of the most fundamental issues every logistic business is facing today. Tracking objects in real time not only requires processing power and storage but also r devices and the techniques for collecting data in real time. To tackle these issues, Musa *et al*. (2014) introduced the concept of embedded devices which process data as a background activity, while others presented the idea of local databases (Li and Xiao, 2013) and warehouses which process their data and send them back to the main server for integration.

Security is the second issue in this type of architecture that includes physical and virtual (data related) threats. Physical security is one of the most common problems which most logistics companies are facing nowadays (Zailani *et al*., 2015) especially in less developed countries. These issues include missing items or boxes from containers or vehicles. Data security is another challenge which many logistics companies are facing in which the most common issues relate to encryption of tags including RFID and QR Code (Hogpracha and Vongpradhip, 2015; Qi *et al*., 2015). Using QR codes, one can find out almost everything about the container as well end user personal information such as name, address, contact number, etc. To prevent unauthorized access to users' personal information, several authors (Musa *et al*., 2014; Li and Xiao, 2013; Hogpracha and Vongpradhip, 2015; Qi *et al*., 2015) proposed encryption algorithm, so no unauthorized person obtains private information without having the key.

The problem of processing Big Data is associated with the data type and the unstructured or scattered nature of data available in a variety of formats is the third issue of this architecture. For example, in logistics businesses, thousands of GPS coordinates, RFID tags and QR Code data are available for tracking and tracing vehicle in real time requiring significant processing power and capabilities which ultimately need resources and are costly. Faster data processing cannot be achieved without optimizing the processing capabilities, storage of data and the way data are collected. To solve this limitations Waller and Fawcett (2013) and Musa *et al*. (2014) and Zhong *et al*. (2015) proposed algorithms, hardware devices and modified software to get better results with lower cost.

Storing large amounts of data in real time is an expensive process and requires high storing capacity that is considered as an issue in this architecture. To solve this issue some authors proposed centralized databases where everything will be stored in a structured database (Oliveira *et al*., 2015), whereas others proposed the concept of local databases to provide ease of data access and data management (Li and Xiao, 2013; Papatheocharous and Gouvas, 2011). After analysis, we found these proposed solutions lack efficiency because storing large amounts of data in the relational database will not only increase processing time but will also be confusing and increase the chances of errors. Another limitation of this proposed solution is that for data analysis purposes through the application of filters requires an enormous amount of time. Another reason why the proposed approaches are not optimal is because they increase data retrieval time if volumes are high (Zailani *et al*., 2015; Au *et al*., 2017; McCreary and Kelly, 2014).

Integrated Track and Trace in Supply Chain based in RFID and GPs

One of the first advanced models was proposed by He *et al*. (2009); the purpose of their framework is to continuously monitor objects in the supply chain. Their model used many IOT based technologies such as RFID, GPS, Web Services and geolocation information so that the object can be a monitored in real time. A GPS is used to gather exact vehicle locations with the help of mobile apps which are installed in the vehicle. The app collects and sends location coordinates information to a central server known as EPCIS Gateway with the help of the GPS and GPRS receivers. This information is stored in an EPCIS gateway database which is accessible with the help of a web interface known as web services. With the help of Application Programming Interface (API), other applications can access and use the stored information for a variety of purposes.

The biggest problem with this proposed solution is the structure of the database. Although, the author introduced separate databases for each party (manufacturer and supplier), to use the structured, centralized database requires complex architecture which is not a good approach to dealing with data from various sources including sensors, etc. The performance of this system is high because the way data are gathered from different sources and combined into a single gateway is good, but the processing time is high due to the centralized structure as no compression techniques have been applied and cost is high regarding the data processing.

Hybrid Cargo-Level Tracking System for Logistics

In this area, one of best work was done by Yang *et al.* (2010) who proposed a hybrid approach to solve tracking and tracing issues for cargo. The vehicle position information is gathered using GPS and wireless sensor networks. The main idea is to propose a low-cost model to perform continuous vehicle monitoring. The whole architecture is divided into three layers, hybrid network, infrastructure and Central Server. Hybrid network infrastructure integrates several technologies including Wi-Fi, GPS, RFID and ZigBee for entities tracking. The intelligent monitoring device is embedded and the core role of this device is to detect motions through ZigBee networks.

Although this approach solves certain limitations of the architecture proposed by He *et al.* (2009) and Jiang *et al.* (2014) regarding processing time, the custom developed hardware cost much more than the other architecture. Also, it is not clear how the proposed architecture performs in real-time tracking since the authors were unable to present how they combined data from different sources and processes.

State of Art Logistics Architecture as Roadmap: eTracer

Papatheocharous and Gouvas (2011) presented a model for tracking cargo in real-time known as eTracer (Fig. 1). The architecture is divided into four components that are Mobile Logistics Stations Network, Fixed Logistics Stations Network, Communication Server and Web Application Server.

This model presented the concept of component-based architecture which divided into four components as shown in Equation 1 (Papatheocharous and Gouvas, 2011). Although proposed solution solves many of the issues of Yang's *et al.* (2010) and He's *et al.* (2009) models by proposing a backend server and dividing it into three parts based on the nature of data. Still, there are certain limitations of Equation 1 in which the most important is custom built in hardware for tracking object reliably:

$$VT = \sum_{k=0}^{n}\left(1 - \frac{I_D}{S_C}\right) + C_H * L_S * Si \tag{1}$$

Where:
I_D = Input data into system
L_S = Local server for storing data
Si = Share information
S_C = Secure object in moving vehicle through devices
C_H = Customer hardware designed for data processing

When the vehicle passes from one station to another, each station receives the data from the installed RFID and sends them to the local server via a network. Then the local server sends the received information to the communication server which ultimately performs communication among different components as shown in Equation 2 (Papatheocharous and Gouvas, 2011). The limitation of this equation is that that data processing is done through custom hardware which can only handle the linear data but not able to handle complex data structures:

$$DP = \sum_{k=0}^{n}\left(I_D + D_P\right) \tag{2}$$

Where:
DP = Data processing
I_D = Input data into system

Moreover, the communication server classifies the information collected from the cargo and stores it in the SQL. Different authorized actors or parties get access to the data in the form of reports, tracking information, etc. through a web application as shown in Equation 3 (Papatheocharous and Gouvas, 2011). The limitation of this equation is that the centralize server is design to store data which must be in linear form and only can be stored in relational database which can be good for certain data format but not for unstructured data type:

$$DS = \left(LS^2 + CS^2\right) * S_D \tag{3}$$

Where:
DS = Data Storage
C_S = Centralized server for storing data
L_S = Local server for storing data
S_D = Store data in structured database

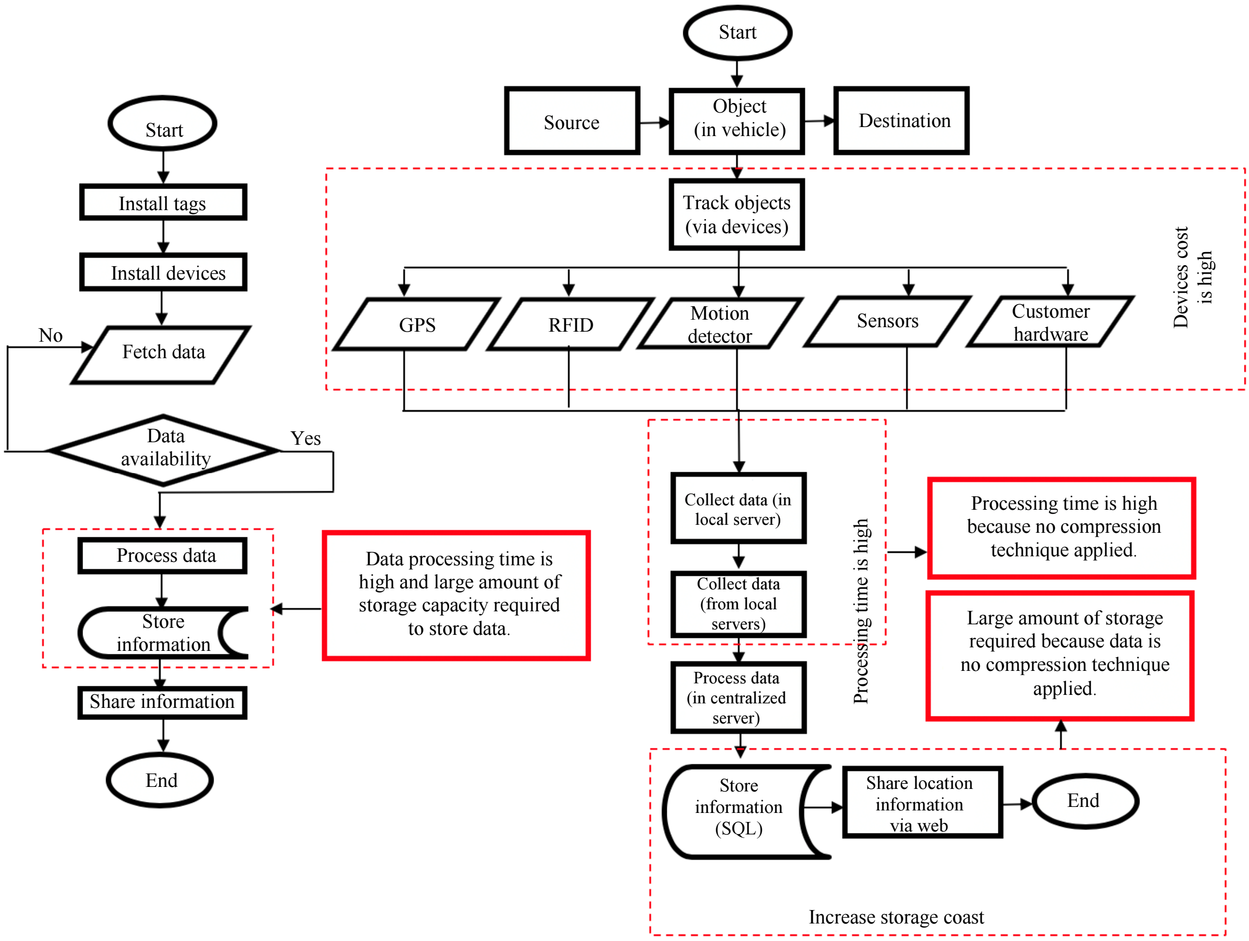
Start
Install tags
Install devices
Fetch data
No
Data availability
Yes
Process data
Store information
Share information
End
Data processing time is high and large amount of storage capacity required to store data.
Start
Source
Object (in vehicle)
Destination
Track objects (via devices)
GPS
RFID
Motion detector
Sensors
Customer hardware
Devices cost is high
Collect data (in local server)
Collect data (from local servers)
Processing time is high
Processing time is high because no compression technique applied.
Large amount of storage required because data is no compression technique applied.
Process data (in centralized server)
Store information (SQL)
Share location information via web
End
Increase storage coast

Fig. 1: Tracking cargo in real-time known as eTracer (Waller and Fawcett, 2013) [The red border refers to the limitation of the state of art model]

Proposed Solution

To overcome the limitations of processing time and data storage problems in current logistics architectures, a Hybrid logistics architecture is proposed. Figure 2 presents the proposed architecture and its components, divided into different layers, data collection, data processing and distributed data storage. All hardware devices, RFID Tags, QR Code and RFID/QR Code reader and scanner, fall under the data collection category. Local servers, a centralized server and the compression algorithm for parsing data fall under Data Processing and all the local databases fall under data processing. The logical flow of the proposed Hybrid Logistics Architecture in Ubiquitous Environment (HLAiUE) can be seen in Fig. 2 and is explained in detail in the following subsections.

Data Processing

Papatheocharous and Gouvas (2011) presented a model which solves many problems of logistics systems including the problem found in Yang *et al.* (2010) and He *et al.* (2009) architectures. In the current solutions, the author (Papatheocharous and Gouvas, 2011) proposed custom build device in which the data is divided into 3 layers based on the nature of data. The custom build device is used to process the data in the background which makes the process fast. However, there are certain limitations of this solution in which the most important are that (a) the solution did not show how much data this device is capable of handling at a time because the data can be fetched through concurrent transactions. If the number of transactions increased at a certain point, the device would be unable to handle that data. Similarly, if the number of transaction decreases, the device will be unable to operate at its full capacity which is a waste of resources due to the cost involved in maintaining it. To solve this limitation, we used distributed data processing and a compression algorithm in our proposed system in a ubiquitous environment which can be seen in Fig. 3.

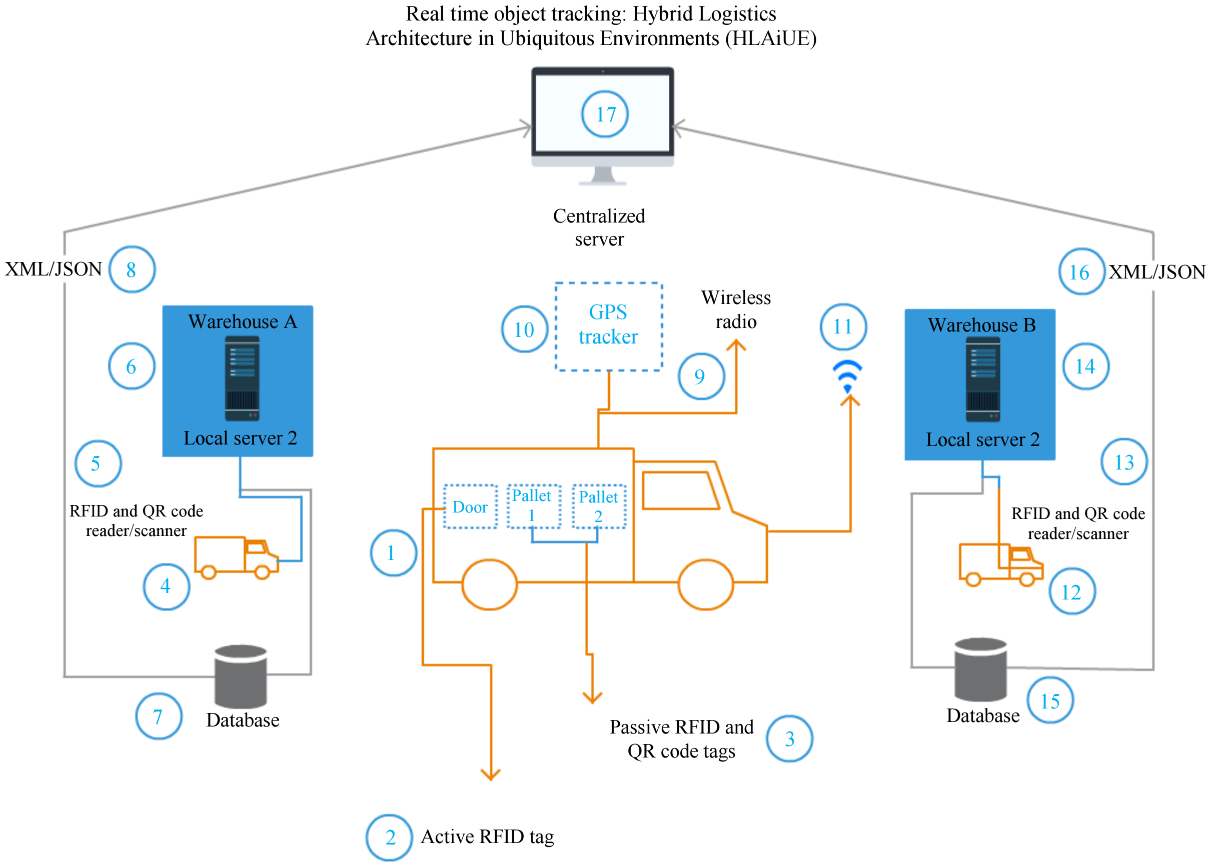

Fig. 2: Proposed Hybrid Logistics Architecture in Ubiquitous Environments (HLAiUE)

Distributed Data Processing

In this part of the proposed architecture, two or more processors involve processing data based on the size of data. The concept of using distributed data processing is to handle and process data dynamically. If the number of concurrent transactions increases, more processors get involved so that the optimum results can be achieved. Similarly, if the number of concurrent transactions is decreasing, fewer processors will be used to produce the same results at less cost. There are many advantages of distributed data processing of which the most important are performance, processing time and flexibility and reliability of the environment. These advantages are explained in the following subsections.

Improvement Performance and Reduced Processing Time

Papatheocharous and Gouvas (2011) only one device handles all the processing which means only a certain amount of data can be processed each time. However, in our proposed architecture, multiple processors are used to process data based on the size of the data with the number of processors increasing dynamically. This will not only reduced data processing time but also improve performance. For example, to track an object in real-time in a moving the vehicle is possible by fetching GPS coordinate data from different devices broken into modules and allocated to different machines where they are processed simultaneously. This significantly reduces processing time and improves performance.

Flexibility

The proposed architecture performed data processing in a distributed environment using Microsoft Azure services, where each computer in a network environment is located in different geographical locations. The benefit of this in our proposed architecture is that each machine will be in a different warehouse and all of them will be connected via the internet and will be able to process data in parallel. This makes our architecture more flexible. Moreover, our architecture is also flexible in terms of increasing or decreasing processing power.

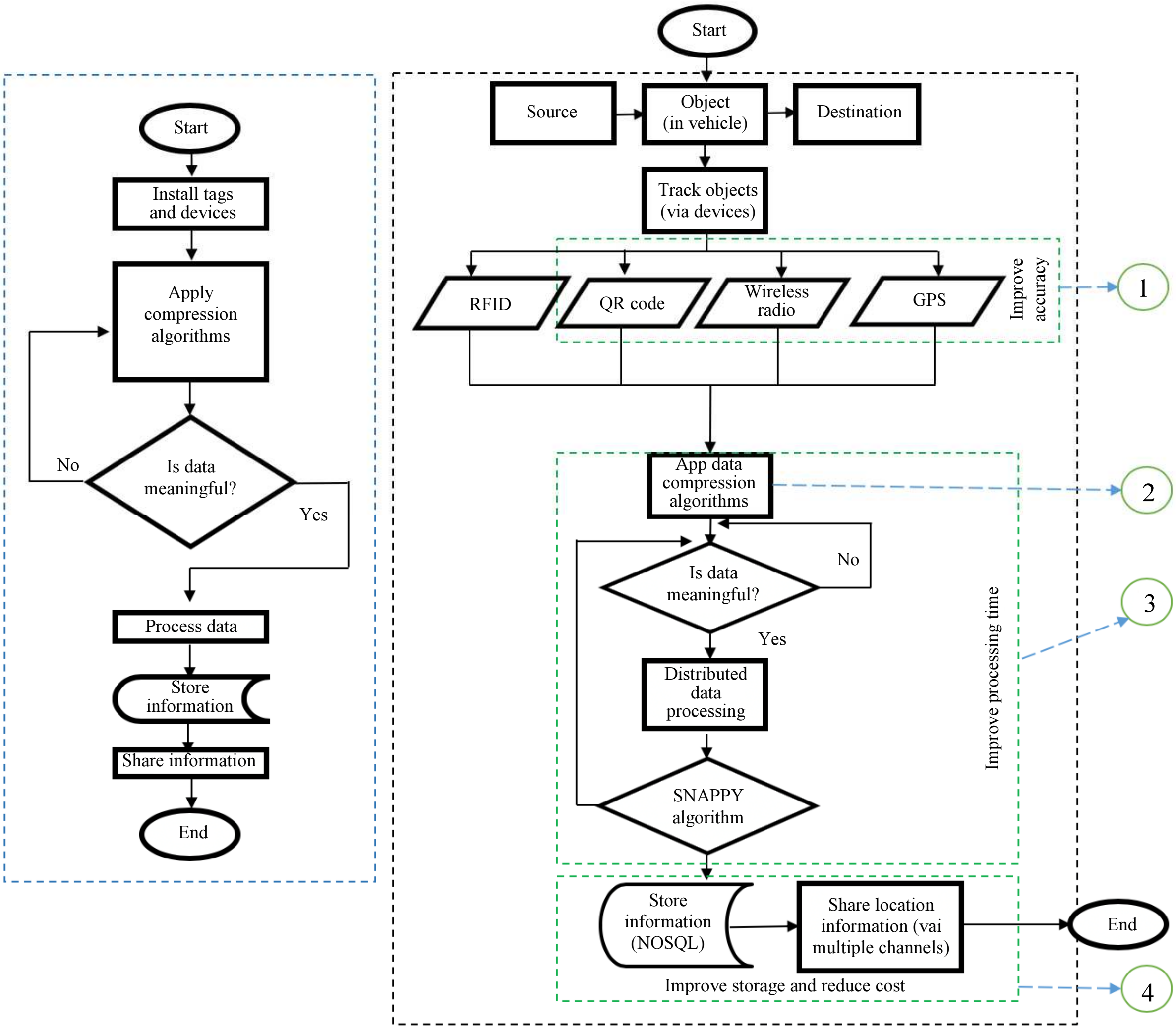

Fig. 3: Flowchart of the proposed architecture for tracking vehicle in real-time [The green borders refer to the new parts in our proposed system. The blue border refers to the logical flow of the model and the black border gives the details of this logical flow]

Reliable

Hardware issues and software bugs can cause single-server processing to malfunction and fail, resulting in a complete system breakdown. For handling these issues, distributed data processing is more reliable, since multiple control centers are spread across different machines. Issues on one machine cannot impact the whole network since another machine in the network can take over its processing capability. This makes our proposed system more reliable and robust compared with the custom built device introduced by Papatheocharous and Gouvas (2011).

Data Compression

In this part of our data processing, we have tested four different compression algorithms and have chosen the best one, the Snappy algorithm chosen because it provides the fastest processing time with high accuracy and an optimum data compression ratio. Comparison results for testing the four algorithms are presented in Table 1. Snappy is a compression/decompression library provided by Google. It aims for maximum compression speed rather than compatibility with other compression libraries. In our proposed logistics architecture, Snappy performs best because its speed of compression is high.

The chosen algorithm, Snappy, is to compress the received data from the GPS receiver. The idea is to reduce the size of the data while maintaining compression speed. The higher the data size, the more processing power and storage is required which increases processing time and cost. The algorithm in Table 1 takes input data from data sets and applies the compression algorithm (Snappy) to improve processing time by reducing the size of the data.

Table 1: Improving processing time algorithm

```
INPUT: Test input data
OUTPUT: Compressed Data
BEGIN
Step 1: Use data from devices (a) as input to HLAiUE
Step 2: INPUT Test GPS data samples I with N number of rows
Step 3: START nested loop
        IF the data is NOT compressed DO
                Repeat step 3
        ELSE
                Successful
RETURN compressed data
END
```

Distributed Database

To improve storage capacity, reduce cost and enhance the speed of accessibility of data in HLAiUE, we used a non-relational database in a distributed environment whereas Papatheocharous (Musa *et al*., 2014; Papatheocharous and Gouvas, 2011) and Li proposed SQL (Li and Xiao, 2013) for their model and architectures. The overall goal is to handle and store data in an efficient and distributed manner so that the data become available on demand to all parties. To achieve this, all collected GPS and RFID data from all devices will be stored in a NOSQL database server known as MongoDB. The idea is to store data in a non-relational manner so that the accessibility of data can become easier and faster. If we compare NOSQL with SQL there are more benefits of using NOSQL over SQL because the volume of data is high and the NOSQL is much better in handling big data and provide the fastest accessibility to data whereas, SQL is good for structured data and handling of small to medium size of data. Beside this, we proposed the NOSQL in a distributed environment, with the concept of using a distributed database being to achieve flexibility so that the database can grow or shrink dynamically without requiring further effort.

Proposed Equation

Input data received from several sources, which ensure the object information receive from time to time. This information then passed to local server so that the data can be processed through custom built hardware device and stored in local server.

Custom hardware device to improve the accuracy of vehicle as shown in Equation 4 (Yang *et al*., 2010):

$$V'T = \sum_{k=0}^{n}\left(1 - \frac{I_D}{S_C} + 1 - L_S\frac{C_H}{D_P}\right) + L_S \qquad (4)$$

Where:
I_D = Input data into system
S_C = Secure object in moving vehicle through devices
L_S = Local server for storing data
C_H = Customer hardware designed for data processing
D_P = Data processing

The Equation 4 has modified by us by adding one new component known as distributed data processing so that all the data must be process in a distributed environment without depending on limited hardware resources and custom-built hardware devices as presented in Equation 5:

$$IVT = V'T * (1 + DP') * (1 - C_H) \qquad (5)$$

Where:
IVT = *Improved Vehicle Tracking*
I_D = Input data into system
C_H = Customer hardware designed for data processing
DP' = Data processing

Data processing is the key element which process the large amount of data in real through custom-built hardware device. The main critical component of their system is the Central Server with several services so that the received data can be processed through central server, this can be depicted in Equation 6 (Yang *et al*., 2010).

The key limitation of custom-built hardware is the capacity to handle complex data structure and the large amount of data at a given time:

$$D'P = \sum_{k=1}^{n}\left(I_D + C_H\right) \qquad (6)$$

Where:
I_D = Input data into system
C_H = Customer hardware designed for data processing

The limitation of Equation 6 solved by us by enhancing data processing in a distributed environment, compressing the data so that large amount of unstructured data can be processed in a short possible time with ow latency rate as shown in equation 7:

$$EDP = D'P = \sum_{k=1}^{\infty}\left(D_C - \frac{I_D}{S_C}\right) \qquad (7)$$

Where:
I_D = Input data into system
C_H = Customer hardware designed for data processing
D_C = Real time data compression in distributed environment

Improved data storage is the key element to store large amount of information, refer to Equation 3 limitation which is S_D, in our proposed architecture, we modified S_D by $S_{D'}$. Instead of storing data in a finite number of data servers, we can now store data in a distributed environment, where the number of servers can be increase or decrease as per the data size. The whole concepted is represented in Equation 8:

$$IDS = \left(L_S^2 + C_S^2\right) * S_{D'} \tag{8}$$

Where:

IDS = Improved Data Storage
C_S = Centralized server for storing data
L_S = Local server for storing data
S_D = Store data in structured database

Experimental Validation and Discussion

This section shows the implementation results of the proposed system implemented in java programming language. A distributed environment has been setup in the cloud via Microsoft Azure using Spark and Hadoop technologies and the NOSQL database server MongoDB. The proposed system has three main defined layers, data collection, distributed data processing and distributed database. In the first layer, Microsoft GeoLife GPS dataset and QR code dataset for location tracking and tracing are used. In the second layer, all the data GPS and QR Code are processed in the Distributed Environment. In this layer, several experiments have been performed using different data compression algorithms with a variety of data. In the third layer, all the processed data has been stored in a de-centralized non-relational database known as NOSQL.

In this study, four well-known data compression algorithms Gzip, Bzip2, Lz4 and Snappy have been tested with GPS data in a distributed environment to improve processing time so that the end user is informed about the location of their product in real-time without any delay. For experimental purposes, 1.5 GB total GPS data was selected which was divided into different chunks. The performance of four compression algorithms namely Gzip, Bzip2, Lz4 and Snappy over reading and writing from/on Hadoop file system and a NoSQL database server MongoDB.

Table 2: GZIP and BZIP4 algorithms results [Execution of the tests were performed in four different cases by varying the number of core processors, RAM and size of data. First case (GZIP and BZIP4), one processor, 2 GB RAM and up to one thousand MB data have been selected]

INPUT					GZIP						BZIP4					
Test case	Proc-essor	Memory (MB)	Data type (Decimal)	Original data Size (MB's)	Compr-essed time (sec)	Decomp-ression Time (sec)	Query time compressed data (sec)	Query time decompressed data (sec)	Compressed data size (MB's)	Compr-ession ratio (%)	Compr-essed time (sec)	Decomp-ression time (sec)	Query time compressed data (sec)	Query time decompressed data (sec)	Compressed data size (MB's)	Compre-ssion Ratio (%)
1	1	2048	D	10	69	9	69	9	2	500%	38	4	38	4	2	500%
	1	2048	D	100	249	2	249	2	22	455%	181	3	181	3	18	556%
	1	2048	D	500	520	4	520	4	41	1220%	310	5	310	5	34	1471%
	1	2048	D	1000	1269	5	1269	5	183	546%	1285	3	1285	3	149	671%
	0	2048	D	1500	Failed						Failed					
2	2	2048	D	10	34.5	5	34.5	5	2	500%	19	2	19	2	2	500%
	2	2048	D	100	125	1	125	1	22	455%	90.5	2	90.5	2	18	556%
	2	2048	D	500	260	2	260	2	41	1220%	155	3	155	3	34	1471%
	2	2048	D	1000	635	3	635	3	183	546%	643	2	643	2	149	671%
	0	2048	D	1500	Failed						Failed					
3	4	2048	D	10	17.3	2	17.3	2	2	500%	9.5	1	9.5	1	2	500%
	4	2048	D	100	62.3	1	62.3	1	22	455%	45.3	1	45.3	1	18	556%
	4	2048	D	500	130	1	130	1	41	1220%	77.5	1	77.5	1	34	1471%
	4	2048	D	1000	317	1	317	1	183	546%	321	1	321	1	149	671%
	0	2048	D	1500	Failed						Failed					
4	4	4096	D	10	8.63	1	8.63	1	2	500%	4.75	1	4.75	1	2	500%
	4	4096	D	100	31.1	0	31.1	0	22	455%	22.6	0	22.6	0	18	556%
	4	4096	D	500	65	1	65	1	41	1220%	38.8	1	38.8	1	34	1471%
	4	4096	D	1000	159	1	159	1	183	546%	161	0	161	0	149	671%
	0	4096	D	1500	Failed						Failed					

Table 3: LZ4 and Snappy algorithms results [Execution of the tests were performed in four different cases by varying the number of core processors, RAM and size of data. Second case (LZ4 and Snappy), one processor, 2 GB RAM and up to one thousand MB data have been selected]

INPUT					LZ4						SNAPPY					
Test case	Proce-ssor	Memory (MB)	Data type (Decimal)	Original data size (MB's)	Compr-essed time (Sec)	Decomp-ression time (sec)	Query time compressed data (sec)	Query time decompressed data (sec)	Compressed Data size (MB's)	Comp-ression ratio (%)	Comp-ressed time (Sec)	Decom-pression time (sec)	Query time compressed data (sec)	Query time Decompressed Data (sec)	Compressed Data size (MB's)	Compr-ession Ratio (%)
1	1	2048	D	10	22	1	22	1	4	250%	23	2	23	2	4	250%
	1	2048	D	100	134	2	134	2	38	263%	137	3	137	3	37	270%
	1	2048	D	500	184	2	184	2	70	714%	192	2	192	2	69	725%
	1	2048	D	1000	1064	2	1064	2	313	319%	982	2	982	2	307	326%
	0	2048	D	1500	Failed						Failed					
2	2	2048	D	10	11	1	11	1	4	250%	12	1	12	1	4	250%
	2	2048	D	100	67	1	67	1	38	263%	69	2	69	2	37	270%
	2	2048	D	500	92	1	92	1	70	714%	96	1	96	1	69	725%
	2	2048	D	1000	532	1	532	1	313	319%	491	1	491	1	307	326%
	0	2048	D	1500	Failed						Failed					
3	4	2048	D	10	5.5	0	5.5	0	4	250%	5.8	1	5.8	1	4	250%
	4	2048	D	100	33.5	1	33.5	1	38	263%	34	1	34	1	37	270%
	4	2048	D	500	46	1	46	1	70	714%	48	1	48	1	69	725%
	4	2048	D	1000	266	1	266	1	313	319%	246	1	246	1	307	326%
	0	2048	D	1500	Failed						Failed					
4	4	4096	D	10	2.75	0	2.75	0	4	250%	2.9	0	2.9	0	4	250%
	4	4096	D	100	16.8	0	16.8	0	38	263%	17	0	17	0	37	270%
	4	4096	D	500	23	0	23	0	70	714%	24	0	24	0	69	725%
	4	4096	D	1000	133	0	133	0	313	319%	123	0	123	0	307	326%
	0	4096	D	1500	Failed						Failed					

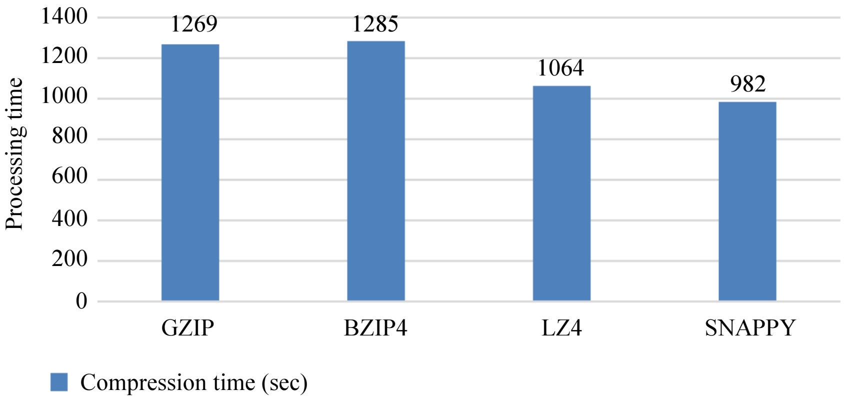

Fig. 4: Time required to process data after applying different compression algorithms

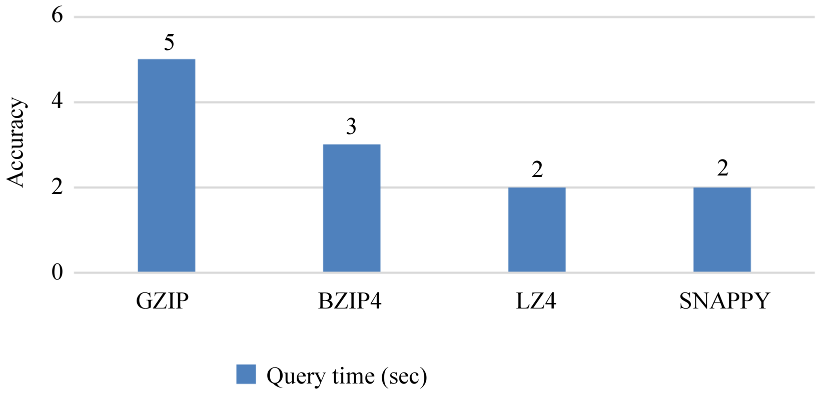

Fig. 5: System accuracy and the time required to run the query

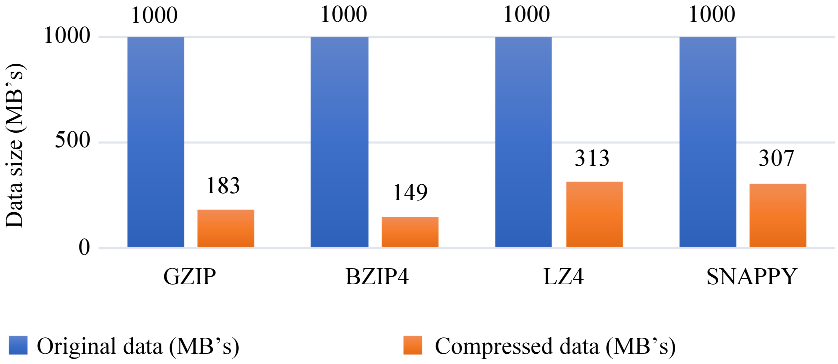

Fig. 6: Data size results before and after compression

As can be seen in Table 1 and 2, the execution of the tests were performed in four different cases by varying the number of core processors, RAM and size of data. In the first case, one processor, 2 GB RAM and up to one thousand MB data have been selected. It can be seen, GZIP and BZIP4 take more time to compress data whereas LZ4 and SNAPPY in the second case take less time to compress the data; this means the processing time of GZIP and BZIP4 algorithms are high (Fig. 4 and 5). But the important thing to note is that the compression ratio for GZIP and BZIP4 are also high, almost 100% and 120% compared with LZ4 and

SNAPPY algorithms. In the other three experiments as shown in Table 2 and 3, after increasing the number of processors from 2 to 4 and the number of RAMS from 2 to 4 GB; the growth ratio remains same after performing 4 different experiments in all cases. If the number of processors increases, the compression time, decompression time and query time decreases whereas compressed data size and compression ratio remain the same as shown in Table 2 case 1.

The second goal of our project was to optimize data size, so that storage capacity can be optimized and the cost can be reduced. During our experiments, we found BZIP4 algorithm performs better than any other algorithm, the reason behind this being that this algorithm works well in compressing data as shown in Fig. 6. To retrieve the data, GZIP algorithm required a lot of processing time which can be seen in Fig. 3, which is not good for our proposed architecture. The only thing is to note when we selected 10 MB data and run LZ4 and SNAPPY algorithms in all other test cases, compression time was slightly low but when the data size of the data increases up to 1000 MB, SNAPPY performs better which is quite clear in our experiments. The system only fails when processors fail, crash or are not available.

Conclusion

In this study, a new secure architecture HLAiUE for logistics in the ubiquitous environment is proposed. It involves collecting a large amount of data from various devices and processing it to get accurate vehicle location in real-time without adding any extra cost or hardware resources. The proposed solution solves three main limitations of current architectures, data processing in real-time provides accurate vehicle location, reduces the size of data for permanent storage and reduces the overall cost. The architecture also includes three main layers, data collection, distributed data processing and distributed data storage layer. For our experiments, we collected and used dataset, processed in a distributed environment using a Microsoft Azure server which not only ensures fast data processing and high accuracy but also saves cost because based on the data size, the number of processors will increase or decrease. Furthermore, our proposed architecture distributes data storage which successfully reduces data size of up to 80% not only saving cost but also providing the fastest accessibility to data so that the vehicle can be tracked in real-time.

Acknowledgement

We are grateful to Mrs. Angelika Maag for proof reading and making corrections to this article. Without her support, it would have not been possible to submit this in the current form.

Author’s Contributions

Abdul Jawwad Patel: He has completed this project as part of his MIT degree program.

Abeer Alsadoon: Worked on the setup of the experiments and gave important suggestions on design of experiments.

P.W.C. Prasad: Made important revisions to most sections of the paper.

A. Elchouemi: Done the technical review of the paper and provided technical guidance. Give the final review and approval for the manuscript to be submitted.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and there are no ethical issues involved.

References

Au, M.H., T.H. Yuen, J.K. Liu, W. Susilo and X. Huang *et al*., 2017. A general framework for secure sharing of personal health records in cloud system. J. Comput. Syst. Sci., 90: 46-62. DOI: 10.1016/j.jcss.2017.03.002

Bandyopadhyay, D. and J. Sen, 2011. Internet of things: Applications and challenges in technology and standardization. Wireless Personal Commun., 58: 49-69. DOI: 10.1007/s11277-011-0288-5

Cao, Q., D.R. Jones and H. Sheng, 2014. Contained nomadic information environments: Technology, organization and environment influences on adoption of hospital RFID patient tracking. Inform. Manage., 51: 225-239. DOI: 10.1016/j.im.2013.11.007

Christou, I.T. and S. Michalakos, 2010. G2G: Location-aware mobile social networking with applications in personalized recommender systems. J. Mobile Multimedia, 6: 73-94.

Chung, K.Y., 2014. Recent trends in convergence and ubiquitous computing. Personal Ubiquitous Comput., 18: 1291-1293. DOI: 10.1007/s00779-013-0743-2

Dawe, R.L., 1994. An investigation of the pace and determination of information technology use in the manufacturing materials logistics system. J. Bus. Logist., 15: 229.

Ghoreishi, S.F. and D.L. Allaire, 2017. Adaptive uncertainty propagation for coupled multidisciplinary systems. AIAA J., 55: 3940-3950. DOI: 10.2514/1.J055893

Giannakis, M., M. Giannakis, M. Louis and M. Louis, 2016. A multi-agent based system with big data processing for enhanced supply chain agility. J. Enterprise Inform. Manage., 29: 706-727. DOI: 10.1108/JEIM-06-2015-0050

Gong, X., S. Gu, Y. Lou, F. Zheng and M. Ge *et al*., 2018. An efficient solution of real-time data processing for multi-GNSS network. J. Geodesy, 92: 797-809.

He, W., E.L. Tan, E.W. Lee and T.Y. Li, 2009. A solution for integrated track and trace in supply chain based on RFID and GPS. Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, Sept. 22-25, IEEE Xplore Press, Mallorca, Spain, pp: 1-6. DOI: 10.1109/ETFA.2009.5347146

Hogpracha, W. and S. Vongpradhip, 2015. Recognition system for QR code on moving car. Proceedings of the 10th International Conference on Computer Science and Education, Jul. 22-24, IEEE Xplore Press, Cambridge, UK, pp: 14-18. DOI: 10.1109/ICCSE.2015.7250210

Imani, M. and U.M., Braga-Neto, 2018. Control of gene regulatory networks with noisy measurements and uncertain inputs. IEEE Trans. Control Netw. Syst., 5: 760-769.

Jiang, J., K.M. Ng, K.L. Poh and K.M. Teo, 2014. Vehicle routing problem with a heterogeneous fleet and time windows. Expert Syst. Applic., 41: 3748-3760. DOI: 10.1016/j.eswa.2013.11.029

Jin, R., G. Chen, A.K.H. Tung, L. Shou and B.C. Ooi, 2018. An optimized iterative semantic compression algorithm and parallel processing for large scale data. KSII Trans. Internet Inform. Syst., 12: 2761-2781.

Leten, B., R. Belderbos and B. Van Looy, 2016. Entry and technological performance in new technology domains: Technological opportunities, technology competition and technological relatedness. J. Manage. Stud., 53: 1257-1291. DOI: 10.1111/joms.12215

Li, M. and Y. Xiao, 2013. Rfid-based tracking and tracing information platform of logistics. Proceedings of the IEEE Conference Anthology, Jan. 1-8, IEEE Xplore Press, China, pp: 1-4. DOI: 10.1109/ANTHOLOGY.2013.6784836

Maity, M. and M. Dass, 2014. Consumer decision-making across modern and traditional channels: E-commerce, m-commerce, in-store. Decis. Support Syst., 61: 34-46. DOI: 10.1016/j.dss.2014.01.008

McCreary, D. and A. Kelly, 2014. Making Sense of NoSQL. 1st Edn., Manning Publications, ISBN-10: 1617291072, pp: 312.

Min, H. and S.J. Joo, 2009. Benchmarking third-party logistics providers using data envelopment analysis: an update. Benchmark.: Int. J., 16: 572-587. DOI: 10.1108/14635770910987814

Musa, A., A. Gunasekaran, Y. Yusuf and A. Abdelazim, 2014. Embedded devices for supply chain applications: Towards hardware integration of disparate technologies. Expert Syst. Applic., 41: 137-155. DOI: 10.1016/j.eswa.2013.07.017

Oliveira, R.R., I.M. Cardoso, J.L. Barbosa, C.A. da Costa and M.P. Prado, 2015. An intelligent model for logistics management based on geofencing algorithms and RFID technology. Expert Syst. Applic., 42: 6082-6097. DOI: 10.1016/j.eswa.2015.04.001

Papatheocharous, E. and P. Gouvas, 2011. eTracer: An innovative near-real time track-and-trace platform. Proceedings of the 15th Panhellenic Conference on Informatics, Sept. 30-Oct. 2, IEEE Xplore Press, Kastonia, Greece, pp: 282-286. DOI: 10.1109/PCI.2011.54

Paris, D.L., M. Bahari and N.A. Iahad, 2016. Business-to-Customer (B2C) electronic commerce: An implementation process view. Proceedings of the 3rd International Conference on Computer and Information Sciences, Aug. 15-17, IEEE Xplore Press, Kuala Lumpur, Malaysia, pp: 19-24. DOI: 10.1109/ICCOINS.2016.7783182

Qi, H., D. Chenjie, Y. Yingbiao and L. Lei, 2015. A new express management system based on encrypted QR code. Proceedings of the 8th International Conference on Intelligent Computation Technology and Automation, Jun. 14-15, IEEE Xplore Press, Nanchang, China, pp: 53-56. DOI: 10.1109/ICICTA.2015.22

Su, X., G. Swart, B. Goetz, B. Oliver and P. Sandoz, 2014. Changing engines in midstream: A java stream computational model for big data processing. Proc. VLDB Endowment, 7: 1343-1354. DOI: 10.14778/2733004.2733007

Valentine, E.L. and G. Stewart, 2013. The emerging role of the board of directors in enterprise business technology governance. Int. J. Disclosure Governance, 10: 346-362. DOI: 10.1057/jdg.2013.11

Vlachos, M., V.G. Vassiliadis, R. Heckel and A. Labbi, 2016. Toward interpretable predictive models in B2B recommender systems. IBM J. Res. Dev., 60: 11:1-11:12. DOI: 10.1147/JRD.2016.2602097

Waller, M.A. and S.E. Fawcett, 2013. Data science, predictive analytics and big data: A revolution that will transform supply chain design and management. J. Bus. Logist., 34: 77-84. DOI: 10.1111/jbl.12010

Weiser, M., 1993. Some computer science issues in ubiquitous computing. Commun. ACM, 36: 75-84. DOI: 10.1145/159544.159617

Yang, G.H., K. Xu and V.O.K. Li, 2010. Hybrid cargo-level tracking system for logistics. Proceedings of the 71st IEEE Vehicular Technology Conference, May 16-19, IEEE Xplore Press, Taipei, Taiwan, p: 1-5. DOI: 10.1109/VETECS.2010.5493655

Yoshii, J. and U. Sumita, 2016. Development of Marketing Flexibility for E-commerce by Assessing Impact of Mobile Devices on Sales with Multiple Classes of Customers. In: Managing Flexibility, Sushil, B.K. and S. Singh (Eds.), Springer India, pp: 273-295.

Zailani, S.H., K. Seva Subaramaniam, M. Iranmanesh and M.R. Shaharudin, 2015. The impact of supply chain security practices on security operational performance among logistics service providers in an emerging economy: Security culture as moderator. Int. J. Phys. Distribut. Logist. Manage., 45: 652-673. DOI: 10.1108/IJPDLM-12-2013-0286

Zhong, R.Y., G.Q. Huang, S. Lan, Q.Y. Dai and X. Chen *et al*., 2015. A big data approach for logistics trajectory discovery from RFID-enabled production data. Int. J. Product. Econom., 165: 260-272. DOI: 10.1016/j.ijpe.2015.02.014