Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com

Review

Towards next-generation heterogeneous mobile data stream mining applications: Opportunities, challenges, and future research directions

Muhammad Habib ur Rehman [a,*], Liew Chee Sun [a], Teh Ying Wah [a,*], Muhammad Khurram Khan [b]

[a] Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
[b] Center of Excellence in Information Assurance, King Saud University, Riyadh, Saudi Arabia

ARTICLE INFO

Keywords:
Frequent pattern mining
Classification
Clustering
Mobile computing
Cloud computing
Edge computing

ABSTRACT

The convergence of Internet of Things (IoTs), mobile computing, cloud computing, edge computing and big data has brought a paradigm shift in computing technologies. New computing systems, application models, and application areas are emerging to handle the massive growth of streaming data in mobile environments such as smartphones, IoTs, body sensor networks, and wearable devices, to name a few. However, the challenge arises about how and where to process the data streams in order to perform analytic operations and uncover useful knowledge patterns. The mobile data stream mining (MDSM) applications involve a number of operations for, 1) data acquisition from heterogeneous data sources, 2) data preprocessing, 3) data fusion, 4) data mining, and 5) knowledge management. This article presents a thorough review of execution platforms for MDSM applications. In addition, a detailed taxonomic discussion of heterogeneous MDSM applications is presented. Moreover, the article presents detailed literature review of methods that are used to handle heterogeneity at application and platform levels. Finally, the gap analysis is articulated and future research directions are presented to develop next-generation MDSM applications.

1. Introduction

The escalation in mobile data was witnessed about 4000-fold over the past decade (Cisco, 2015). Cisco, the big name in network infrastructures, predicts that mobile data will grow up to 30.6 Exabytes (*i.e.* 30.6 billion Gigabytes) by the year 2020 (Cisco, 2015). This massive amount of data will be generated by next generation of mobile systems such as mobile IoTs, WSNs, BSNs, robotics, unmanned aerial vehicles, and satellite systems to name a few (Rehman et al., 2016a). Considering this growth, mobile data will challenge the storage and processing capacities of existing computing systems. Next-generation applications will be developed to handle the data in streaming mode and on-the-fly using in-memory data processing architectures before storing in large scale distributed systems (Zhang et al., 2015). These trends will highlight the importance of data stream mining applications which perform in-memory analytic operations over streaming data in order to uncover hidden knowledge patterns (Krishnaswamy et al., 2012). These knowledge patterns will help understanding the underlying data and benefit in decision making in personal and commercial applications.

Mobile streaming data which is the subset of overall big data is helpful in improving business operations across the enterprises (Rehman et al., 2016b). For example, the analysis of mobile data streams generated by remote vehicles help in optimizing supply chain management operations (Kargupta, 2016). Similarly, the mobile data streams collected from remote customers is useful for creating personalized services for online shoppers (Tan et al., 2016). The governments can also improve the daily and emergency response management operations by analyzing real-time mobile streaming data from citizen's mobile devices (Murphy, 2016). Despite wide applicability, it is quite challenging to decide about where and when to process the streaming mobile data.

This article presents a thorough literature review of existing MDSM applications and platforms in order to establish the state of the art and find the future research directions. A few relevant literature reviews were proposed in the past, however, they emphasized on other perspectives. For example, the authors in Gaber et al. (2005), Parthasarathy et al. (2007), Goel et al. (2010), Fuqiang (2011), Krishnaswamy et al. (2012), Tsai et al. (2014), Gaber et al. (2014a), Nguyen et al. (2015) and Chen et al. (2015) focused on general MDSM

* Corresponding author.
Email addresses: mhrehman@siswa.um.edu.my (M.H.u. Rehman); csliew@um.edu.my (L.C. Sun); tehyw@um.edu.my (T.Y. Wah)

http://dx.doi.org/10.1016/j.jnca.2016.11.031
Received 26 July 2016; Received in revised form 13 November 2016; Accepted 28 November 2016
Available online xxx

algorithms and lack the discussion on application-level and platform-level issues. Similarly, in our previous study (Rehman et al., 2015), we studied mobile data mining applications in batch mode execution and static datasets. To the best of our knowledge, this is the first article that presents the review of MDSM applications and platforms in MECC environments. The article is structured as follows. Section 2 presents the bibliometric analysis of mobile data mining and mobile data stream mining publications which were indexed in web of science databases. Section 3 presents a detailed discussion on execution platforms for MDSM applications and the associated opportunities and challenges. Section 4 presents the taxonomy of heterogeneous MDSM applications. Section 5 presents a thorough literature review of methods for handling heterogeneity in MDSM applications. Section 6 discusses the heterogeneity issues at platform level and Section 7 presents a detailed literature review of selected platforms for MDSM applications. Section 8 presents the gap analysis of existing literature and discusses the future research directions. Finally, the article concludes in Section 9.

2. Bibliometric analysis of Web of Science databases

Research on mobile data mining is growing rapidly in recent years. We performed a preliminary study on Web of Science (WoS) databases (Web of science databases, 2016) by querying the string "mobile data mining". According to retrieved statistics, as of 28th January 2016, the WoS databases indexed 1930 publications in last 27 years (from 1990 to 28th January 2016) from International Scientific Indexing (ISI)-listed journals, conferences and workshop proceedings, and magazines (See Fig. 1). There was no significant research on the topic from 1990 to 2002. Since Year 2002, the miniaturization of technologies and on-board sensing technologies had geared-up the research on mobile data mining. However, the major boom started from Year 2007 when both Google (Android (operating system), 2016) and Apple (Apple iphone history, 2016) released their mobile operating systems.

According to Fig. 1, the number of publications rapidly increased till 2015 which shows that mobile data mining is continuously becoming a hot research topic. In near future, we perceive a major shift towards the research on mobile data mining due to rapid growth in far-edge mobile devices for example smartphones, wearable devices, mobile IoTs, and body sensor networks to name a few. The citation trends for the topic "mobile data mining" are depicted in Fig. 2. The citation analysis showed that publications on the topic of mobile data mining obtained 9041 total citations from 8180 citing publications which were indexed in WoS databases. The popularity of research on mobile data mining is witnessed by the fact that 7935 citing publications were published without self-citations by the respective authors. The average citations per publication is 4.68 with h-index as 40. Fig. 2 also depicts

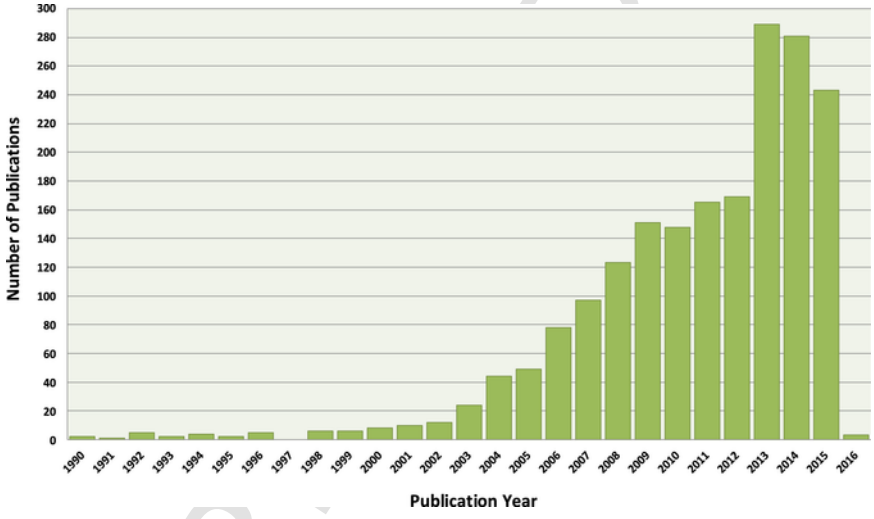

Fig. 1. Year-wise publications (1990–2016).

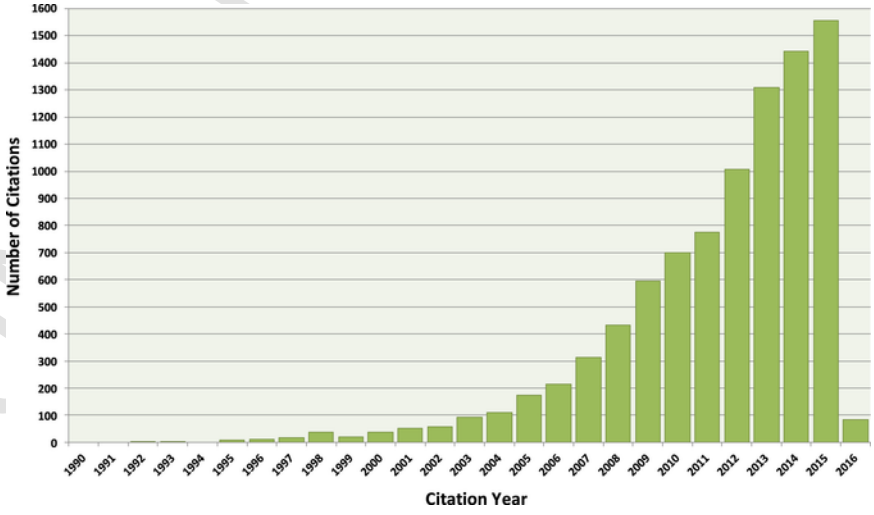

Fig. 2. Year-wise citations (1990–2016).

that arrival of mobile operating systems in 2007, boomed the research on mobile data mining and it is still increasing day by day.

Since the main focus of this article is on mobile streaming data, therefore, we further analyzed the bibliographic records from WoS databases with another query string as "mobile data stream mining". We found 112 publications indexed by WoS databases from Year 1990 to 28th January 2016. These 112 publications were cited by 343 other publications in WoS databases whereby 331 publications do not contain any self-citation by respective authors. The average citation per publication is 3.06 with h-index as 11 which was lower when compared with bibliometric analysis of "mobile data mining" because less number of publications on the topic. Likewise, the major boom in "mobile data stream mining" was also witnessed after Year 2007 and it is rapidly growing. Considering the fast growth of research in MDSM algorithms, applications, and execution platforms, a thorough literature review is presented in this article.

3. Mobile data stream mining platforms

The MDSM platforms facilitate in efficient execution of analytic components. The literature review reveals that MDSM platforms (see Fig. 3) were deployed in multiple topological settings (Abdallah et al., 2015; Gaber et al., 2014b; Haghighi et al., 2013; Jayaraman et al., 2014a). The underlying communication models include multiple computing devices and systems having different form factors. These devices and systems include mobile devices, Internet, and intranet based application servers and cloud data centers to name a few (Jayaraman et al., 2014b; Kargupta et al., 2010; Mukherji et al., 2014). The topological settings of MDSM platforms that are presented in this article are based on far-edge mobile devices, far-edge to far-edge communication models, mobile and immobile edge servers based communication models, mobile cloud computing and mobile edge cloud computing systems.

3.1. Far-edge mobile devices

Far-edge mobile devices are defined as any portable system or device with wireless communication interfaces and ability to produce or process data. Smartphones, wearable sensors, wireless body sensor networks, smart vehicles, and Mobile Internet of Things (IoTs) are a few examples of far-edge mobile devices. Although modern far-edge mobile devices enable rich MDSM applications such as virtual reality, computer vision, and multimedia applications using cloud augmented computational resources (Satyanarayanan et al., 2015) however the execution of heterogeneous MDSM applications inside far-edge devices is a challenging task (Rehman et al., 2015). Far-edge mobile devices usually contain limited computational resources and battery power, therefore, MDSM applications consider these limitations for efficient process execution in mobile environments (Krishnaswamy et al., 2012). Data stream mining components, as shown in Fig. 4, are designed to be light-weight to unleash the maximum utilization of on-board computational resources (Haghighi et al., 2013).

Opportunities: The deployment of MDSM applications in far-edge devices offers multi-fold opportunities. The MDSM applications help in reducing outgoing data streams which in turn reduce network traffic as well minimize the cost of communication in terms of bandwidth utilization and GSM data plans (Jayaraman et al., 2014b). In addition, the close proximity of data sources and computational components in far-edge devices lowers the latency in execution time when compared with offloading raw data streams in external environments such as servers, cloud data centers, and grid computing resources (Jayaraman et al., 2014a). The privacy preservation and local knowledge availability are additional benefits of the deployment of MDSM applications in far-edge devices (Arunkumar et al., 2015). The knowledge patterns acquired after onboard execution of MDSM applications enable local knowledge availability, reduce dependency on external systems for data processing, and preserve the privacy of users' personal data.

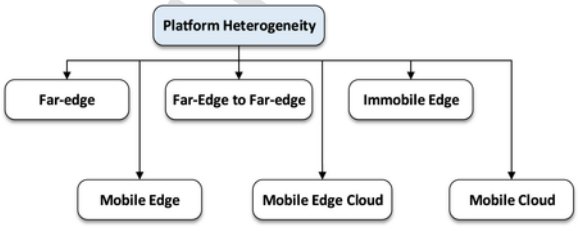

Fig. 3. Taxonomy of MDSM platforms.

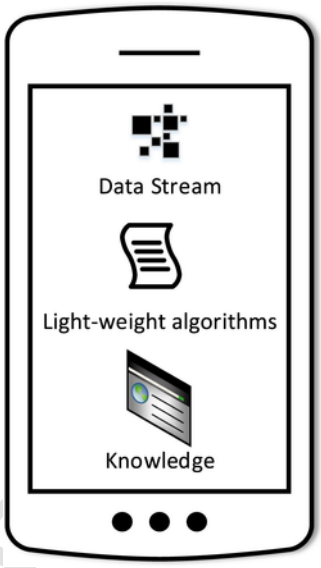

Fig. 4. MDSM applications in far-edge mobile devices.

3.2. Far-edge to Far-edge

The Far-edge to Far-edge (F2F) communication models are based on a set of Far-edge devices that can communicate with each other directly without any additional controlling mechanism or data communication point. For example, F2F communication model (see Fig. 5) facilitates in a direct communication between smart watch like Samsung Gear and a smartphone such as Samsung Galaxy S5 (Samsung unveils galaxy s5 and new gear range, 2014). Similarly, multiple devices owned by a single user such as wearable devices, smartphones, tablet PCs, and laptops can offer a direct communication network through Bluetooth communication

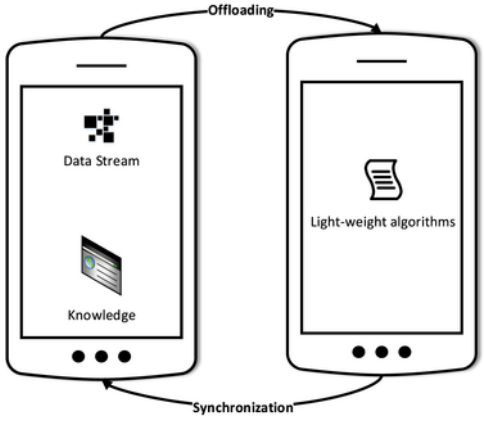

Fig. 5. MDSM applications in F2F communication model.

interfaces and do not require any other communication point such as Wi-Fi router or local wireless hub (Gaber et al., 2014b). The F2F communication model is adequate for single-user multi-device settings where far-edge devices can initiate point-to-point and group communication sessions to execute MDSM applications collaboratively (Framework, 2015).

Opportunities: In F2F settings, the closer far-edge mobile devices can pool the computational resources to augment the resource-constrained far-edge mobile devices with maximum execution support within Personal Area Network (PAN) (Wang et al., 2012; Rehman et al., 2015). In addition, the F2F settings enable to distribute application logic among different far-edge mobile devices for seamless application execution (Li et al., 2015; Wang et al., 2012). For example, the far-edge mobile devices with minute computational facilities perform data acquisition operations and facilitate in data transfer operations in relatively high-power far-edge mobile devices. The high-power far-edge mobile devices execute MDSM application components and synchronize the knowledge among other far-edge mobile devices in PAN (Gaber et al., 2014b).

3.3. Mobile edge servers

Mobile Edge server is defined as any mobile device or mobile system that resides at a one-hop wireless distance from far-edge mobile devices. A mobile edge server enables data stream mining functionality by providing mobile services to thin and thick far-edge mobile devices (*see Fig.* 6). *The thin far-edge mobile devices function as data acquisition and data transfer elements, however, thick far-edge devices enable extra functionality of light-weight data stream mining algorithms*. Some examples of mobile edge servers include frequently co-located far-edge devices such as personal mobile devices (wearable devices, smartphones, Tablet PCs, and laptop computers), far-edge mobile devices owned by co-workers, family members, and friends, and shared far-edge mobile devices such as appliances in smart home environments, and office equipment in smart co-working spaces.

Opportunities: The co-location and co-movement of far-edge mobile devices and mobile edge servers reduce dependency over large-scale centralized systems (Kargupta et al., 2010). In addition, far-edge mobile devices can offload resource-intensive tasks to mobile edge servers without Internet connections by utilizing local communication channels such as wireless hub, Wi-Fi direct, and Bluetooth Low Energy interfaces. Mobile edge servers may own and control by different users, therefore, MDSM applications should be device-centric and mobile edge servers sho-uld provide complete application clones to reduce the high coupling. An added advantage of mobile edge servers is the elastic service availability where far-edge device can offload data mining tasks in multiple mobile edge servers using device-centric task scheduling schemes (Rehman et al., 2016c). The addition of location aware context features in MDSM applications can enable mobile distributed intelligence where multiple far-edge mobile devices can sense and log the data and act as both far-edge mobile devices and mobile edge servers.

3.4. Immobile edge servers

Immobile edge servers are defined as the physically static and resourceful computing systems that reside at a one-hop wireless distance from far-edge mobile devices. The immobile edge servers include cloudlets, micro data centers, radio access network (RAN) servers in GSM networks, application servers, and smart-routers in local area networks to name a few (Bonomi et al., 2012; Satyanarayanan et al., 2015; Bahl, 2015; Ha and Satyanarayanan, 2015). Similar to mobile edge servers, the communication model (see Fig. 6) facilitates thin/thick far-edge devices but physically bounded nature of immobile edge servers enforce collaborative execution models between far-edge mobile devices and immobile edge servers (Ferreira et al., 2010). The collaborative execution model needs to perform operational monitoring at far-edge mobile devices and immobile edge servers for seamless execution of MDSM applications (Sherchan et al., 2012).

Opportunities: The deployment of MDSM applications at immobile edge servers helps in prolonging battery lifetime of far-edge mobile devices (Satyanarayanan et al., 2015). In addition, the availability of high computational resources reduces the application processing time hence minimizes the latency (Bahl, 2015).

3.5. Mobile cloud computing system

Mobile cloud computing (*MCC*) *systems are defined as the computing systems that provide heterogeneous computing, networking, and storage services to far-edge mobile devices through large scale data centers*. The application models for mobile cloud computing based data stream mining applications involve thin and thick far-edge mobile devices (Altomare et al., 2013) (see Fig. 7). For example, wearable devices directly upload data stream in cloud data centers and data stream mining operations are performed in cloud environments. Alternately, far-edge mobile devices, such as in the case of CARDAP, perform data stream mining operations locally using on-board computational resources and enable on-demand data offloading when required (Jayaraman et al., 2014a).

Opportunities: The MCC systems offer many opportunities to augment MDSM applications. The MCC systems enable the provision of highly available and hypothetically unlimited computing, networking, and storage resources through large-scale data centers. The MCC systems enable multiple forms of services namely Storage-as-a-Services (SaaS), Application-as-a-Services (AaaS), Network-as-a-Services (NaaS),

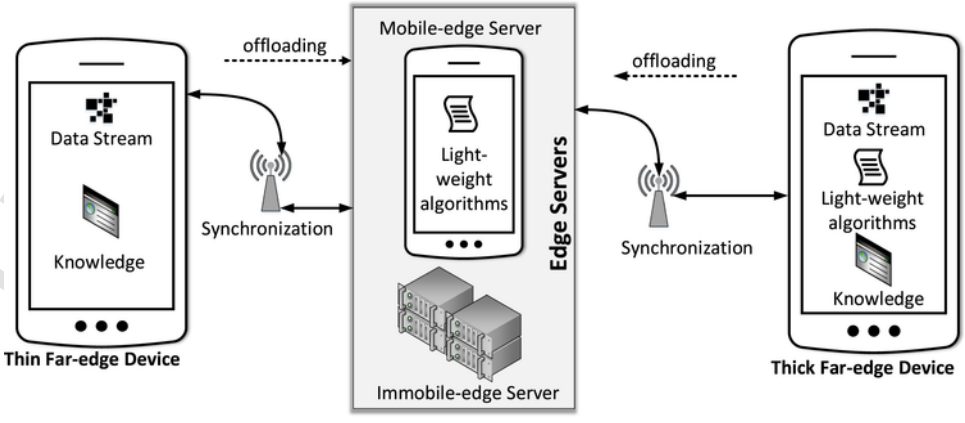

Fig. 6. MDSM applications in edge servers.

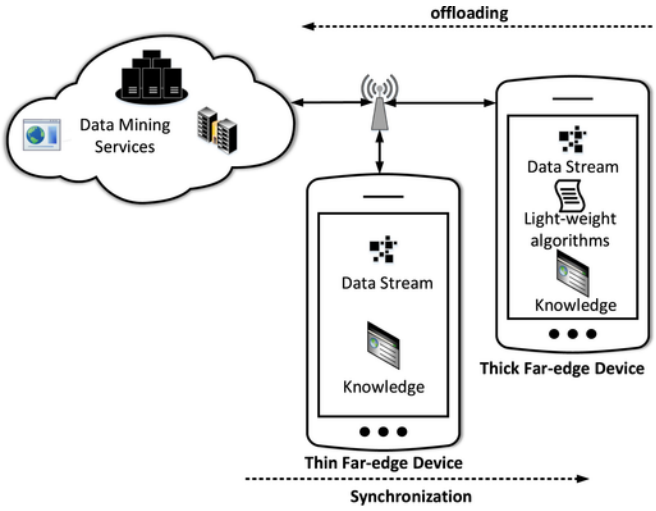

Fig. 7. MDSM applications in mobile cloud systems.

and a large plethora of services at hardware, operating systems, and application levels (Fernando et al., 2013; Sharma et al., 2016).

3.6. Mobile edge cloud computing system

Mobile edge cloud computing (*MECC*) *systems extend the traditional MCC services on the edge of the Internet through mobile and immobile edge servers that reside at one-hop wireless distances from mobile devices*. The MECC systems enable distributed MDSM applications by replication of traditional infrastructure based cloud services in edge servers as well as application partitioning at multiple levels (Ye et al., 2012). The MECC based MDSM applications span over far-edge mobile devices, edge servers, and traditional cloud computing infrastructures (Ha et al., 2014) (see Fig. 8).

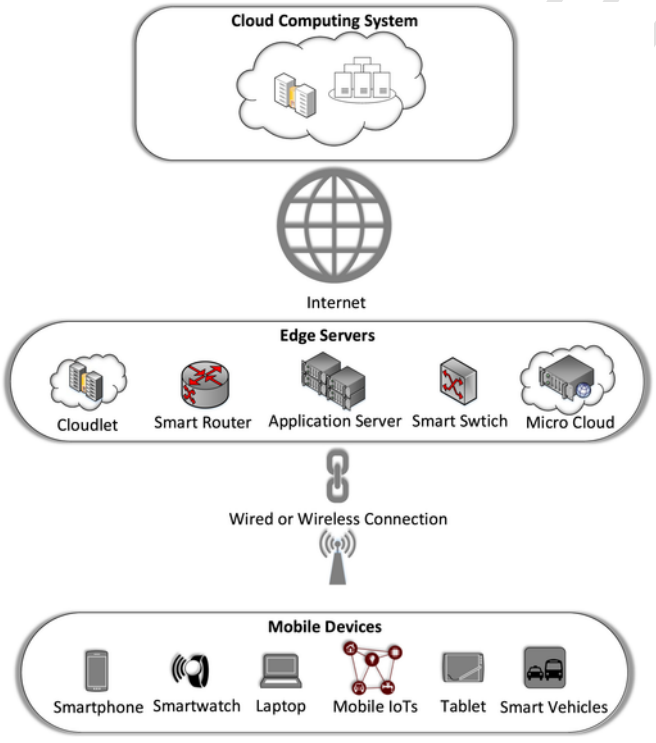

Fig. 8. MDSM applications in mobile edge cloud systems.

Opportunities: The MECC systems provide the scalable computing infrastructure which can help in the deployment of highly distributed MDSM applications (Ye et al., 2012). Far-edge mobile devices in MECC systems perform single-site and multiple-site computation offloading (Simoens et al., 2013). In addition, the unlimited computational and storage support from traditional infrastructure based cloud computing systems enable to deploy and dedicate heterogeneous resources for edge servers (Ortiz et al., 2015). The edge servers can further utilize the acquired resources for seamless application execution. Edge servers also perform the resource intensive computations to prolong battery life time and minimize latency in MDSM applications (Drolia et al., 2013). Furthermore, the MDSM applications are geographically distributed to minimize the load-balancing efforts in infrastructure based cloud (Luan et al., 2015).

3.7. Challenges

The MDSM platforms need to address several challenges for efficient application execution.

3.7.1. Resource constraints and light-weight data processing

The limitations in battery power, CPU, and memory are the main bottlenecks in far-edge mobile devices, F2F communication models, and mobile edge servers (Krishnaswamy et al., 2012; Gaber et al., 2014a; Rehman et al., 2014). The challenge arises due to miniaturization of computational elements and the constraints of designing small size, light-weight, and less heat dissipating far-edge mobile devices. Since far-edge devices offer limited computational and battery power resources. Existing MDSM platforms adapt the execution behavior according to resource availability and situation awareness which enforce light-weight execution of application components and result in compromising the quality of knowledge patterns (Haghighi et al., 2013).

3.7.2. Compute-intensive operations

Far-edge mobile devices produce continuous data streams therefore MDSM application need to process or store whole data streams in order to uncover maximum knowledge patterns. Although modern far-edge devices come with sophisticated computational elements and enable power saving functions, the heterogeneity in MDSM applications increase the computational complexities of application components. Handling the increased computational complexities together with high data rates is still a challenging task (Gaber et al., 2014a; Rehman et al., 2014).

3.7.3. Distributed application logic

The distribution of application logic among far-edge devices, edge servers, and cloud data centers is a major challenge (Wang et al., 2012). The MDSM applications need to be carefully designed to run the resource-intensive components in relevantly high-power far-edge devices or cloud servers in order to avoid resource unavailability in low-power far-edge mobile devices (Min and Cho, 2011). The application logic could be distributed statically by deploying application components across far-edge mobile devices, edge servers, and cloud servers. The static distributions may introduce high coupling among application components and the applications may fail in F2F and mobile edge servers settings due to unavailability of computational and battery power resources (Braojos et al., 2014; Liu et al., 2013). To handle this issue, the application components could be distributed dynamically or adaptively, however, existing literature still lacks the relevant studies.

3.7.4. Mobility

Far-edge mobile devices constantly move among different communication networks and switches between Wi-Fi, Blue tooth, and GSM-based Internet connections. Keeping a track record of mobility patterns

for seamless application execution is a challenging task especially when far-edge mobile devices continuously and rapidly switches among different communication interfaces (Ahmad and Ahmad, 2016). The mobility becomes a major challenge when MDSM platforms operate in F2F settings or the applications are executed using mobile edge servers. The mobility of devices may also impact the privacy and security of device data because far-edge mobile devices may need to upload data streams to different mobile edge servers (Khan, 2015). In addition, heterogeneity in operating systems of mobile edge servers, programming environments, and communication interfaces requires extensive profiling of mobile edge servers to provide optimal user experience.

3.7.5. Resource provisioning in MECC

The MCC and MECC communication models provide virtual machines (VMs) and containers (i.e. light-weight VMs) for application execution on the edge and cloud servers (Ahmad et al., 2015). Due to fast mobility of devices and continuously streaming data, live VM migration becomes very challenging because the time taken during migration must remained lower than the time spent by far-edge mobile device in the same communication network. The containers enable fast provisioning of cloud resources however it requires a lot of programming efforts to design containers for each MDSM application. In addition with VM migration, the saving and resumption of application states also becomes challenging especially when the far-edge mobile devices continuously switch among mobile edge servers (Ha and Satyanarayanan, 2015).

3.7.6. Dependency over internet connections

Far-edge mobile devices need persistent Internet connections for efficient application execution using MCC and MECC communication models. To handle the connectivity issues, MDSM applications perform onboard data management operations which may quickly hamper onboard memory resources and result in application failure. Therefore, MDSM applications must reduce dependency over Internet connection either by executing application components locally in far-edge mobile devices or by optimizing onboard data management schemes (Sherchan et al., 2012; Jayaraman et al., 2014a).

3.7.7. Increased data communication and high latency

The continuous data production in far-edge mobile device increases the network traffic between far-edge devices and edge servers and cloud servers. In addition, large transfer of raw data stream increases in-network data communication in cloud data centers. The increased data communication results in high bandwidth utilization cost and extra energy consumption for data transfer, data management, and data processing in MCC and MECC systems (Ha and Satyanarayanan, 2015). The cloud servers in MCC systems reside at multi-hop distance from far-edge mobile devices which results in increased makespan in MDSM applications hence increases latency. In addition, high data rates increases the size of data stream which impacts the data communication cost in MDSM applications (Ha and Satyanarayanan, 2015).

In this section, we presented the execution platforms for MDSM applications and discussed the relevant opportunities and challenges. However, MDSM applications in itself need to deal with heterogeneous components. We present a detailed taxonomic discussion on heterogeneous MDSM applications in next section.

4. Heterogeneity in MDSM applications

MDSM applications work in five steps: (a) mobile applications provide functionality to acquire data streams from one or more data sources, (b) fusion of data stream from multiple sources results in information rich data stream representing multiple facets of each data tuple, (c) preprocessing operations enable to improve the quality of data stream by handling missing values, removing noise, and detecting anomalies and outliers, (d) data stream mining operations are performed for online knowledge discovery using different model-based and model-less data mining algorithms, and (e) uncovered knowledge patterns are summarized, integrated and managed for further utilization using multiple knowledge management approaches. Fig. 9 presents the taxonomy of heterogeneous MDSM applications.

4.1. Heterogeneity in data acquisition

Data acquisition in MDSM applications is a challenging task because of massive heterogeneity in multiple aspects. Although data streams are represented as subset of big data, however, it also need to handle few big data properties such as volume, velocity, variety, variability, and veracity.

4.1.1. Volume (Size)

MDSM applications need to handle continuous and unboun-ded data streams, therefore, limiting the size of data stream is a tedious task (Krishnaswamy et al., 2012). MDSM applications handle volume using few methods based on sliding windows and segmentation (Oneto et al., 2015; Abdallah et al., 2015; Wu et al., 2013). The sliding windows are used to sample a preset number of tuples at a given time in-

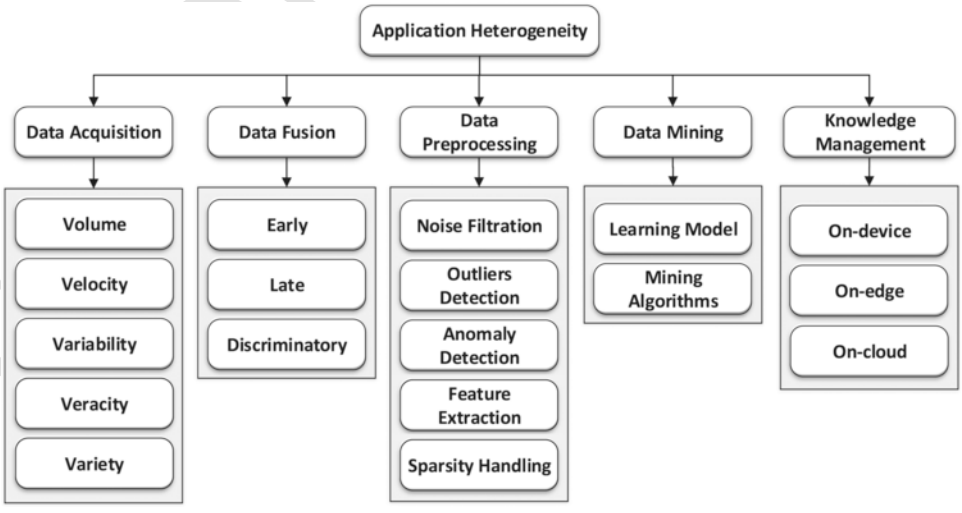

Fig. 9. Taxonomy of heterogeneous MDSM applications.

terval. The size of sliding windows may vary in different applications. Sliding windows are used in two modes. The overlapping sliding windows contain a portion of data which overlaps the previous window. The overlapping is performed to improve the quality of data stream so that the useful data items on the start and end of the windows should not be wasted. The non-overlapping sliding windows also play a vital role in some of the application areas. For example, mobile health applications with non-overlapping sliding windows are more useful than overlapping windows. Similarly, data segmentation is used as an alternate of sliding windows methods where the buffered data streams are equally distributed in a finite number of chunks for lateral processing.

4.1.2. Velocity (speed)

The speeds of incoming data streams play a vital role in MDSM applications (Gaber et al., 2009). Velocity is the key challenge in mobile applications that increases latency. MDSM applications handle velocity in two ways: (a) the applications collect raw data in centralized data stores for lateral data processing and (b) the data is analyzed using in-memory operations before data storage. In the first approach, MDSM application create a delay between data acquisition and knowledge discovery. This strategy is more useful for analysis of historical data. The second approach is more appropriate for real-time data analysis. However, in this case, MDSM applications compromise on the quality of knowledge patterns because the continuous entrance of data stream bounds to one-time data processing (Gama, 2013).

4.1.3. Variety (number and type of data sources)

The variety property represents collection of the data stream from heterogeneous data sources and multiple data formats (*i.e.* structured, unstructured, and semi-structured) (Rehman et al., 2015; Swan, 2012). MDSM applications collect the data stream from multiple data sources including the on-board sensors (such as in IoT systems, wearable devices, and smartphones) and off-board sensors such as accumulating data from other devices or external environments. A thorough review of the data sources is presented in Rehman et al. (2015) for interested readers.

4.1.4. Variability (variable data production rates)

The data rates in MDSM applications vary according to the nature of data sources and application requirements. Therefore, variability property of data stream needs serious attention in MDSM applications in order to deal with inconsistencies and uncertainties of incoming data streams. In addition, MDSM applications sometimes need to handle consistently continuous data streams and sometimes data streams come in episodic patterns. This behavior increases the importance of variability property of data streams.

4.1.5. Veracity (authenticity of data sources)

The veracity property shows that MDSM application need to collect the data streams from trustworthy and reliable data sources. The veracity property ensures that the data streams are collected in an authentic way and the correctness of the data is guaranteed. Therefore, if properly handled, the veracity property of the data stream improves the quality and usefulness of collected data. Otherwise, inefficient handling of veracity property may lead to degradation of quality of knowledge patterns produced by MDSM applications.

4.2. Heterogeneity in data fusion

Data sources generate data stream with different sampling frequencies and introduce heterogeneity in data fusion operations. For example, the sampling frequency of accelerometer is absolutely different when compared with a parallel data stream that is being sampled from the camera.

4.2.1. Early data fusion

Early data fusion methods are applied when raw sensor data from multiple data sources is sampled at the same data rate which is measured as a number of samples in each given time period (Oneto et al., 2015; Mukherji et al., 2014; Khan et al., 2013; Wang et al., 2012). For example, activity recognition applications that are sampling data streams from accelerometer and GPS location sensor at the same time with same sampling rate. The average sampling frequency of accelerometer for activity recognition applications is recommended as 25HZ however user location do not change so frequently hence produce a lot of redundant GPS data. Similarly, if the sampling frequency is set as 1HZ the under-sampling of accelerometer produce inaccurate data hence affects the results of data mining algorithms. Therefore early data fusion strategies are helpful for MDSM applications with low sampling rates but under perform in case of high variance in sampling rates of different data sources.

4.2.2. Late data fusion

Late data fusion methods are applied after preprocessing the data stream (Min and Cho, 2011; Sherchan et al., 2012; Jayaraman et al., 2014a). The late fusion strategies helps in addressing the data redundancy issues. The data stream from multiple data sources is sampled at different sampling rates, preprocessed and the resultant data is integrated to generate events data streams. For example, the accelerometer samples the sensors at 25 Hz while the GPS is sampled at 1 Hz. The late data fusion strategies first create sliding windows of 25 readings from the accelerometer and performs the feature extraction from each sliding window. The extracted features and GPS locations are integrated and transformed into events. When compared with early data fusion, the late data fusion strategies helps in data reduction and improving data quality.

4.2.3. Discriminatory features based data fusion

Far-edge mobile devices such as wireless sensor networks and mobile IoTs may involve homogeneous sensing settings where multiple data sources represent same information (Shoaib et al., 2014). However, sensor configurations and placement may affect in quality data acquisition. The discriminatory fusion methodologies involve the identification of quality data sources and fusion of discriminatory features which may help in improving the quality of uncovered knowledge patterns.

4.3. Heterogeneity in data preprocessing

The preprocessing operations enable to improve the quality of the data stream. The heterogeneity in preprocessing operations arise when MDSM applications need to handle missing values, remove noise, and detect anomalies and outliers from the data stream.

4.3.1. Noise filtration

Noise refers to the inclusion of extraneous and irrelevant information in mobile data streams (Khan et al., 2010). The data streams becomes noisy due to multiple reasons such as improper placement of sensors, wrong sensor configurations, and inducement of environmental noise among others.

4.3.2. Outliers detection

Outliers refer to misreported data points where the acquired data streams do not fully represent the desired data streams. Numerous classification and clustering methods are used to detect and remove the outliers (Hromic et al., 2015).

4.3.3. Anomaly detection

Anomaly detection refers to the presence of anomalous data points in acquired data streams (Suarez-Tangil et al., 2015). The anomaly detection helps in improving the quality of knowledge patterns.

4.3.4. Feature extraction

Massive data streams need to handle efficiently. The feature extraction methods help in extracting features (also known as attributes) from incoming data streams (Siirtola and Roning, 2013; Yang et al., 2014; Oshin et al., 2015). Feature extraction methods convert data streams from unstructured and semi-structured formats into structured data formats.

4.3.5. Sparsity handling

Highly sparse data may hamper the performance of far-edge mobile devices in some cases (Wang et al., 2013). Similarly, low sparsity also degrades the performance of data stream mining applications. Therefore, handling sparsity and maintaining an adequate level of sparsity in data stream mining applications help in improving the quality of knowledge patterns.

4.4. Heterogeneity in data stream mining

Data stream mining algorithms vary in terms of frequent pattern mining, classification, and clustering schemes and the learning models vary in terms of supervised, unsupervised, semi-supervised, and deep learning schemes.

4.4.1. Learning model heterogeneity

The learning models represent the machine learning algorithms and used to support the clustering, classification, and frequent pattern mining algorithms for knowledge discovery. The heterogeneity in learning model arises in terms of learning type, learning model, and learning modalities. The training type varies in terms of supervised, unsupervised, semi-supervised, and deep learning models. The supervised learning models are trained using labeled data streams wherein the learning models adopt the recognition behaviors in order to predict and classify the future data streams (Cord and Cunningham, 2008; Dogan and Tanrikulu, 2013). On the contrary, the unsupervised learning models are trained without labeling the data streams wherein the learning model adopt the behavior and group the future events on the basis of similarities and dissimilarity measures (Huang et al., 2014). The semi-supervised learning models are initially trained with labeled data streams, however, it adopts with unlabeled data for future recognition (Settles, 2012; Goldberg et al., 2009; Triguero et al., 2015). The deep learning models are the multi-level implementations of supervised, semi-supervised, and unsupervised learning models wherein data streams are segregated on the basis of preset criteria set by application designers and separate learning models are developed for each subspace of the data stream (Martens, 2010).

MDSM applications train learning models either online or offline. The online learning models are trained inside far-edge mobile devices, edge servers, and cloud data centers using live data streams. However offline learning models are trained using already collected data (Liang et al., 2014). Although online learning models are computationally complex, the knowledge patterns produced by online learning models are more accurate and can cater the evolving data streams (Gomes et al., 2012a). Alternatively, the offline learning models produce less accuracy and become personalization-agnostic because of training with historical data (Khan et al., 2013).

4.4.2. Mining algorithm heterogeneity

MDSM algorithms are categorized as classification, clustering, and frequent pattern mining algorithms.

Classification: The classification algorithms use supervised, semi-supervised, and deep learning models in order to classify the input data streams. The classifiers use single class recognition or multi class prediction models depending upon the application requirements. The classification algorithms vary in terms of (a) universal model, wherein a global model is used for the whole data stream; (b) personalized model, wherein the local models are used depending upon the needs of users, applications, and machines; and (c) adaptive model, wherein the classification process starts from a global model which is retrained as a personalized model (Lu et al., 2012). Despite various modeling techniques, the classification algorithms posses the data labeling overhead either manually or automatically, therefore, the automation of classification algorithms is a laborious and time consuming process.

Clustering: The clustering algorithms use unsupervised learning models and cluster data points on the basis of similarities and dissimilarities (Abdallah et al., 2015; Haghighi et al., 2013; Suarez-Tangil et al., 2015). The measurement of similarities and dissimilarities depends on cluster centroids and the attribute values of data points. The data clustering algorithms in MDSM applications vary in terms of subspace clustering, density based, centroid-based, hierarchical, subspace, spectral, and constrained based methods. The choice of these techniques solely depends upon the type and nature of data to be clustered as well as the application requirements. However, clustering algorithms are not widely adopted in far-edge mobile device-based data stream mining systems due to high and sometimes unlimited computational requirements.

Frequent Pattern Mining: The Frequent pattern mining algorithms are applied over similar sets of items (Agrawal et al., 1994; Rehman et al., 2014). The frequent pattern mining algorithms mines the frequently occurring itemsets with a preset frequency threshold named as minimum support (*minsup*). The frequent itemsets are further mined to find the associations among itemsets and establish the association rules among them. The rule establishment is performed using another threshold called minimum confidence (*minconf*). The itemsets and their association rules vary in simple, closed, maximal, rare, sporadic and utility based itemsets. These algorithms are generally designed to mine only frequent patterns and/or to find associations among different itemsets. Overall research in frequent pattern mining varies from basic patterns to multilevel and multidimensional patterns, to extended patterns for data sets and streams.

4.5. Heterogeneity in knowledge management

The integration, storage, and utilization of knowledge patterns in MDSM applications take place at various places.

4.5.1. On-device

The on-board storage refers to the storage capabilities of far-edge devices that are used to store locally uncovered knowledge patterns (Wang et al., 2013; Yoon, 2013). In addition, the synchronized knowledge patterns for personalized user experience are also stored on-board far-edge mobile devices.

4.5.2. on-edge

The service provision from edge servers enables data reduction (Ye et al., 2012; Yoon, 2013). The location-aware aggregation of knowledge patterns facilitate in reduced data transfer in remote environments and minimize bandwidth utilization.

4.5.3. Remote

Conventionally knowledge patterns are integrated and stored in remote data stores which include cloud data center, clusters, grids, and application servers. Remote knowledge aggregation is useful for global knowledge discovery (Ferreira et al., 2010).

In this section, we presented the detailed taxonomic discussion on heterogeneous MDSM applications. In the next section, we present a thorough literature review of proposed methods for heterogeneous MDSM applications.

5. Handling heterogeneity in MDSM applications

5.1. Methods for handling data acquisition heterogeneity

Number and type of data sources vary depending upon the nature of data stream mining systems. The application specific systems facilitate only essential data sources, however, the number of data sources in generic systems varies. For example, the application specific systems such as mobile activity recognition system mostly use accelerometers, GPS receivers, and magnetometers (Oneto et al., 2015; Khan et al., 2013; Yang et al., 2014). Alternately, generic systems like CARDAP, OMM, and MobiSens caters bundles of sensory and non-sensory data sources to enable generality and support wide range of applications (Jayaraman et al., 2014a; Haghighi et al., 2013; Wu et al., 2013). The data sources include homogeneous and heterogeneous type of data sources. Homogeneous data sources are mainly used when same type of data is produced by multiple data sources such as multiple accelerometers deployed in wireless body sensor networks (Shoaib et al., 2014). Alternately heterogeneous data sources are used when MDSM applications need to collect and analyze data stream from different data sources. The heterogeneous data sources produce integrated and multi-dimensional data stream. Systems such as MineFleet and MobiSens utilizes heterogeneous data sources and produce multi-format information-rich data streams (Kargupta et al., 2010; Wu et al., 2013).

The data streams are collected from both on-board and off-board data sources (Pasricha et al., 2015; Abdallah et al., 2015). Similarly, the systems are designed as first-person data stream mining systems whereby personal data is analyzed and personalized knowledge discovery is performed (Gu et al., 2011; Mukherji et al., 2014). Alternatively, the data stream mining systems integrate the data streams from multiple users/devices/data sources for the production of generalized knowledge patterns (Pasricha et al., 2015; Jayaraman et al., 2014b). MDSM systems handle multiple data types ranging from numerical and textual data to multimedia and event data streams. Literature review reveals that most of the systems cater only the numerical data streams such as accelerometer axis and GPS coordinates, however, a few systems such as MSM (Mukherji et al., 2014) and OMM (Haghighi et al., 2013) supports multiple data formats. These data types finally lead towards the nature of data streams as structured, unstructured, and semi-structured data tuples.

To handle the resource constraints, MDSM systems adopt different data collection strategies which differ in terms of collection mode, and amount and nature of collected data. The data streams are either collected offline for lateral data processing or immediately processed using either on-board computational resource, offloaded to other computational system/infrastructures such as edge servers, cloud servers, or perform collaborative data processing by harnessing computational resources from nearer similar devices/systems. MDSM systems either collected raw data streams or initially process and reduce the data streams to lower on-board resource consumption as well as bandwidth utilization cost for data offloading.

In addition, some studies reported the representative and context aware data collections strategies as well. The representative data collection strategies are useful when multiple data sources generate same data stream representing the same knowledge. The representative data collection strategies work best in crowd-sensing like application scenario and useful in handling highly redundant data streams. The contextual information about user states, locations, and behavior helps in inferring current situations of users which in turn facilitate in data reduction whereby data stream mining applications only collect the data stream when a specific situation occurs. CAROMM utilizes context aware data collection strategies based on fuzzy situation inference model which infer current situation of users (Sherchan et al., 2012). Table 1 presents the detailed literature review of methodologies for handling data acquisition used by selected studies.

5.2. Methods for handling data fusion heterogeneity

Literature review reveals that early data fusion is adopted in data stream mining systems which collect data streams from multiple data sources and aggregate for further processing (Srinivasan et al., 2014; Braojos et al., 2014). Early data fusion results in redundant and noisy data streams therefore introduce inefficiency and extraneous resource consumption in mobile devices. A few studies use late data fusion strategies whereby collected data streams are preprocessed in parallel before data fusion (Min and Cho, 2011; Jayaraman et al., 2014a). The late data fusion strategies consume onboard computational resources however it improves the data quality for lateral data processing. Late data fusion is useful when preprocessed data is integrated from multiple persons and different data sources. Although discriminatory data fusion strategies are also proposed by the researchers but existing literature still lacks its application in MDSM systems (Shoaib et al., 2014).

Data fusion strategies either work as online methods where all computations are performed in memory or work offline where data streams are stored onboard before data fusion (Oshin et al., 2015; Yoon, 2013). The online strategies are effective and improve system performance in terms of latency and local storage I/O operations. However, in-memory computations sometimes result in data loss and reduced data quality when dealing with large and complex data streams. Offline data fusion facilitates in improved data quality and complete data streams however quickly hampers onboard storage resources. Table 2 presents a detailed literature review of data fusion heterogeneity in MDSM applications.

5.3. Methods for handling data preprocessing heterogeneity

MDSM applications adopt various data preprocessing methods for sliding windows based data stream segmentations, feature extraction, data conversion from unstructured to structured formats, signal analysis, noise and data filtration, privacy and security, dimension reduction, outliers' detection and many others.

The selection of preprocessing methods depend upon the nature of data streams and application requirements. For example, overlapping sliding windows based segmentations are used for activity recognition applications (Suarez-Tangil et al., 2015). Similarly, anonymization and encryption techniques facilitate in privacy and security features of MDSM applications (Mukherji et al., 2014).

Similar to data fusion operation, data preprocessing operations are performed in offline and online mode (Lu et al., 2012; Yuan and Herbert, 2014). The offline preprocessing methods are applied over historical data which is acquired and stored using onboard storage. The online data preprocessing operations are performed in memory. However, in-memory computations become challenging due to variant complexities of data preprocessing algorithms. Table 3 presents the detailed literature review of preprocessing methods.

Table 1
Data acquisition heterogeneity.

Reference	Data Sources	Types	Name	Users	Data Types	Nature	Mode
Oneto et al. (2015)	2	Off-board	Accelerometer and Magnetometer	Multiple	Numeric/Textual	Structured	Offline
Pasricha et al. (2015)	1	Onboard	Application Log Files	NA	Textual	Structured	Online
Abdallah et al. (2015)	72	Onboard	Accelerometer	Multiple	Numeric/Textual	Structured	Offline
Suarez-Tangil et al. (2015)	Numerous	Onboard	Sequence of System Calls	NA	Textual	Structured	Online
Boukhechba et al. (2015)	1	Onboard	GPS Receiver	Multiple	Textual	Structured	Offline
Haghighi et al. (2013)	2	Off-board	ECG Sensors, Accelerometers	Single	Numerical	Structured	Offline
Gomes et al. (2012b)	1	Onboard	Accelerometer	Multiple	Numerical	Structured	Online
Liu et al. (2012)	1	Off-board	Accelerometers	Multiple	Numerical	Structured	Online
Khan et al. (2010)	5	Onboard	Accelerometer	Multiple	Numerical	Structured	Offline
Mukherji et al. (2014)	3	Onboard	Application Log Files, Call Records, Location	Single	Textual	Structured	Offline
Abdallah et al. (2012)	72	Onboard	Accelerometer	Single	Numerical/Textual	Structured	Offline
Sidek et al. (2014)	Numerous	Off-board	ECG Sensors	Multiple	Continuous Signals	Unstructured	Offline
Khan et al. (2013)	5	Onboard	Accelerometer	Multiple	Numerical	Structured	Offline
Srinivasan et al (2014)	3	Onboard	Application Log Files, Call Records, Locations	Multiple	Textual	Structured	Offline
Siirtola and Roning (2013)	1	Onboard	Accelerometer	Multiple	Numerical	Structured	Offline
Siirtola and Röning (2012)	1	Onboard	Accelerometer	Multiple	Numerical	Structured	Offline
Yang et al. (2014)	1	Onboard	Accelerometer	Multiple	Numerical	Structured	Offline
Lu et al (2012)	1	Onboard	Microphone	Multiple	Audio	Unstructured	Offline
Donohoo et al. (2014)	Numerous	Onboard	GPS/user Interactions	Multiple	Numerical/Textual	Structured	Offline
Oshin et al. (2015)	1	Onboard	Accelerometer	Multiple	Numerical	Structured	Offline
Rai et al. (2012)	1	Onboard	Accelerometer	Multiple	Numerical	Structured	Offline
Wang et al. (2012)	7	Off-board	5 Accelerometers and 2 RFID	Multiple	Numerical	Structured	Offline
Gaber et al. (2014b)	Numerous	Both	Multiple Data Sources	Multiple	Both	Both	Both
Ortiz et al. (2015)	1	Onboard	Camera	Multiple	Images	Unstructured	Offline
Braojos et al. (2014)	9	Both	Accelerometer	Multiple	Numerical	Structured	Offline
Min and Cho (2011)	Numerous	Both	Accelerometer and Magnetometer	Multiple	Numerical	Structured	Offline
Stahl et al. (2012)	Numerous	Both	Multiple Data Sources	Multiple	Both	Both	Both
Jayaraman et al. (2014b)	13	Both	Multiple	Multiple	Both	Both	Online
Wu et al. (2013)	8	Onboard	Multiple	Multiple	Both	Both	Offline
Sherchan et al. (2012)	Numerous	Both	Multiple Data Sources	Multiple	Both	Both	Offline
Jayaraman et al. (2014a)	Numerous	Onboard	Multiple Data Sources	Multiple	Both	Both	Online
Lin et al. (2013)	1	Onboard	GPS Receiver	Multiple	Numerical	Structured	Offline
Yuan and Herbert (2014)	2	Both	Accelerometer and Gyroscope	Multiple	Numerical	Structured	Both
Talia and Trunfio (2010)	Numerous	NA	Numerous	Multiple	NA	NA	Offline
Kargupta et al. (2010)	Numerous	Onboard	On-board Vehicle Sensors	Multiple	Both	Both	Online
Yoon (2013)	2	Onboard	Accelerometer and GPS	Multiple	Numerical	Structured	Online
Gu et al. (2011)	2	Off-board	Accelerometer and Camera	Multiple	Both	Both	Offline

5.4. Methods for handling data mining heterogeneity

MDSM applications use different learning models based on supervised, unsupervised, semi-supervised and deep learning approaches. Currently, most of the learning models are trained offline in desktop PCs, servers, or cloud systems. Some studies trained learning models in mobile devices as well however online training of learning models inside mobile environments is a challenging task. The challenge arises because training types of supervised, unsupervised, semi-supervised and deep learning approaches differ. In the case of supervised learning models, the training data stream needs to be labeled/annotated so that learning models can accurately recognize and predict the future similar data streams.

However, the labeling of data streams differs in manual, automatic, and observational settings. The manual labeling is performed when

Table 2
Data fusion heterogeneity.

Reference	Nature of Fused Data	Data Fusion	Fusion Mode
Oneto et al. (2015)	Raw	Early	Offline
Abdallah et al. (2015)	Raw	Early	Offline
Haghighi et al. (2013)	Raw	Early	Offline
Khan et al. (2010)	Raw	Early	Offline
Mukherji et al. (2014)	Raw	Early	Offline
Abdallah et al. (2012)	Raw	Early	Offline
Sidek et al. (2014)	Preprocessed	Early	Offline
Khan et al. (2013)	Raw	Early	Offline
Srinivasan et al (2014)	Raw	Early	Offline
Donohoo et al. (2014)	Raw	Early	Offline
Oshin et al. (2015)	Raw	Early	Offline
Rai et al. (2012)	Raw	Both	Offline
Wang et al. (2012)	Raw	Early	Offline
Gaber et al. (2014b)	Raw	Both	Online
Ortiz et al. (2015)	Raw	Early	Online
Braojos et al. (2014)	Raw	Early	Online
Min and Cho (2011)	Preprocessed	Late	Online
Stahl et al. (2012)	Raw	Both	Online
Jayaraman et al. (2014b)	Raw	Early	Online
Wu et al. (2013)	Raw	Early	Offline
Sherchan et al. (2012)	Preprocessed	Late	Offline
Jayaraman et al. (2014a)	Raw	Late	Online
Lin et al. (2013)	Raw	Early	Online
Yuan and Herbert (2014)	Raw	Early	Online
Talia and Trunfio (2010)	Raw	Early	Offline
Kargupta et al. (2010)	Raw	Early	Online
Yoon (2013)	Raw	Both	Online
Gu et al. (2011)	Raw	Early	Offline

each segment/chunk of the data stream is manually annotated however this process is quite laborious and needs a lot of efforts. An alternate methodology is the adoption of automatic application driven labeling where the applications are configured at the time of data collection and the resultant data streams are annotated accordingly. The automatic labeling is more promising as compared to manual labeling in order to reduce the training efforts. The observational settings further enhance the automatic labeling by allowing users to intervene in data labeling process. In this approach the learning models are initially trained in automatic settings however in the case of discrepancies users are allowed to intervene by manually labeling the data streams.

The selection of learning algorithms significantly impacts the performance of MDSM applications in order to perform energy-efficient, cost-effective, highly accurate data stream mining operations. For deployment in mobile environments, the internal structures of learning models and their processing behavior play an important role in devising the computational complexity of learning models. In essence, MDSM applications need to perform online data stream mining operations on continuous data streams. Therefore, most of the studies either separate the training and recognition processes or use shallow data structures like arrays, lists, or pruned trees for improved efficiency. Learning in MDSM applications is performed to achieve multiple objectives which include system level and application level performance enhancements. The system level performance objectives include battery life enhancements in mobile devices and performing offloading decisions in mobile cloud settings. However the majority of methods used learning models to enhance application performance in terms of change detection from uncertain data streams, model personalization, prediction and optimization of next locations, online activity recognition, finding emerging patterns, to name a few.

Once the learning models are trained and deployed, the MDSM applications process the incoming data streams in both online and offline mode. The offline data streams are stored in the onboard local storage and processed whenever the feasible environment for data stream processing is available. The online data streams are directly processed either using on-board computational resources or offloaded in other devices and systems in F2F, mobile-edge, MCC, or MECC settings. Majority of the studies in literature used classification algorithms due to low computational complexities and easy deployment as compared to clustering and frequent pattern mining algorithms. The classification algorithms are used for multiple purposes that include onboard classifications, on-wireless node classification, distributed classification, multi-level classification, and light-weight classification. A few studies implemented light-weight clustering and association rule mining algorithms which show the practicality of clustering and frequent pattern mining algorithms in mobile environments. Table 4 presents a detailed literature review of data stream mining heterogeneity in MDSM applications.

5.5. Methods for handling knowledge management heterogeneity

Since MDSM applications process data streams at multiple devices and systems, therefore, the integration and summarization of knowledge patterns needs careful attention. MDSM applications usually run the knowledge discovery operations such as learning and recognition and knowledge management operations such as integration, summarization, and storage of knowledge patterns at the same device or system. However, few studies present the synchronization/transfer of knowledge patterns among different systems whereby the knowledge patterns are stored either in local storage such as onboard data stores in far-edge mobile devices or in remote data stores such as those in cloud data centers and edge servers. The hierarchical knowledge management facilitate in enabling both local and remote storage settings. Hierarchical knowledge management strategies enable local storage at a lower level where far-edge mobile devices manage the knowledge patterns using on-board settings. At the second level, multiple devices transfer the knowledge patterns to nearer edge servers which integrate and manage local data stores. Finally, multiple edge servers in different geographical settings transfer the knowledge patterns to centralized cloud data centers which enable knowledge integration for a global view.

Knowledge visualization is another challenge that MDSM applications need to handle efficiently. MDSM applications provide the visualization functionalities either on-screen in far-edge mobile devices or provide a web interface for remote visualization. On-screen visualization in far-edge mobile devices is handy for real-time applications however limited screen size and energy intensive operations quickly hampers the on-board computational and battery resources. The knowledge management strategies work in both online and offline mode. The online knowledge management strategies integrate, summarize and visualize the knowledge patterns before storage and lateral aggregation if required. However, offline strategies first integrate, summarize, and store the knowledge patterns. In such cases, the on-demand visualization is enabled whereby knowledge patterns are visualized when required. For example, the historical activity patterns of a user or noise level in a particular city. Table 5 presents the detailed literature review of knowledge management heterogeneity in MDSM applications.

In this section, we presented a detailed literature review of heterogeneous MDSM applications. In the next section, we present a thorough

Table 3
Data preprocessing heterogeneity.

Reference	Data Preprocessing Method	Type of Preprocessing Algorithm	Mode	Preprocessing Objective
Oneto et al. (2015)	Sliding Windowing with 50% Overlap	Time and Frequency Domain Feature Extraction	Offline	Extraction of 561 Features
Abdallah et al. (2015)	Clustering Sliding Windows	KNN Clustering	Online	Extraction of Features from Clusters and Sub-clusters
Suarez-Tangil et al. (2015)	Sliding Windowing with 50% Overlap	Histogram Features	Online / Offline	Feature Extraction for Anomaly Detection
Haghighi et al. (2013)	ECG signals converted using mobile health open source framework	ECG signals to numeric value conversion	Offline	Feature Extraction
Khan et al. (2010)	Feature Extraction Methods	Noise Filtering and Feature Extraction	Offline	Feature Extraction from Non-linear Space
Abdallah et al. (2012)	Clustering of Sliding Windows	Cluster-based Features	Online	Extracted Features from Clusters
Mukherji et al. (2014)	Anonymization and Encryption	Privacy and Security	Offline	User De-identification
Sidek et al. (2014)	QRS Selection/Normalization	Feature Selection and Normalization	Offline	Feature Extraction from ECG Data
Khan et al. (2013)	SMA, LDA, and KDA	Noise Filtering and Feature Extraction	Offline	Feature Extraction from Non-linear Space
Srinivasan et al (2014)	Anonymization and Encryption	Privacy and Security	Offline	User De-identification
Siirtola and Roning (2013)	Feature Extraction	Statistical Feature Extraction Methods	Offline	Features Extraction from Accelerometer Data
Siirtola and Röning (2012)	Feature Extraction	Statistical Feature Extraction Methods	Offline	Features Extraction from Accelerometer Data
Yang et al. (2014)	Feature Extraction	Time and Frequency Domain Features	Offline	Feature Extraction
Lu et al (2012)	Feature Extraction	Statistical and Acoustic Features	Offline	Feature Extraction from Voice Data
Donohoo et al. (2014)	Principle Component Analysis	Feature Extraction	Offline	Feature Extraction Onboard Sensors
Oshin et al. (2015)	Feature Extraction	Mathematical Functions for Feature Extraction	Offline	Feature Extraction from Accelerometer Data
Rai et al. (2012)	Feature Extraction	Statistical Feature Extractions	Offline	Feature Extraction
Wang et al. (2012)	Dynamic Time Wrapping	Template Matching	Offline	Template Matching
Gaber et al. (2014b)	Numerous	Numerous	Both	Multiple
Ortiz et al. (2015)	Sift/Surb/ORB	Feature Extraction Method	Online	Feature Extraction
Braojos et al. (2014)	Time-domain and Frequency domain	Feature Extraction Method	Online	Feature Extraction
Min and Cho (2011)	Segmentation	Activity-based Classification	Offline	Segmentation
Stahl et al. (2012)	Numerous	Numerous	Both	Multiple
Jayaraman et al. (2014b)	Sliding Windows with 50% Overlap	FFT and Light-weight Analysis	Online	Multiple
Wu et al. (2013)	Sliding Windowing for Segmentation	NA	NA	NA
Sherchan et al. (2012)	Change Detection	Light-weight Clustering	Online	Quality Data Collection
Jayaraman et al. (2014a)	Light-weight Algorithms	Light-weight Clustering	Online	Quality Data Collection
Yuan and Herbert (2014)	Sliding Windowing with 50% Overlap and Feature Extraction	66 Time Domain and Frequency Domain Features Extracted through Statistical Methods	Online	Multi-user Data Collection
Yoon (2013)	Filtration methods are applied	Filtration	Online	Data Filtration
Gu et al. (2011)	Sliding Windows based Segmentation	NA	Offline	Improving Data Quality

literature review of methods that are used to handle platform level heterogeneity for MDSM applications.

6. Handling heterogeneity in data stream mining platforms

The heterogeneous devices and systems offer variable computational and energy resources to MDSM applications. Therefore, platform level heterogeneity is handled using adaptation methods, application partitioning and computation offloading schemes, and data transfer strategies.

6.1. Adaptation

The adaptations are made at system-level to adapt the generic processing behavior of data stream mining applications. Alternately, the adaptation strategies work at algorithm level by altering the execution behaviors of data stream mining algorithms. The adaptations are made using multiple parameters such as data rate, memory, CPU, context aware features, learning models, and specific event. The data rate

Table 4
Data stream mining heterogeneity.

Reference	Learning Mode	Learning Type	Learning Algorithm	Learning Objective	Mining Mode	Learning Model	Training Mode	Data Mining Algorithm	Purpose
Oneto et al. (2015)	Offline	SL	Multiple Learning Algorithms	Battery Life Enhancement	Online	Yes	Offline	Feed Forward Selection	Classification
Pasricha et al. (2015)	Online	SL	Q-learning	Battery Life Enhancement	Online	Yes	Offline/Adaptive	Bayesian Classifier	Classification
Abdallah et al. (2015)	Both	SL/UL	Incremental/Active Learning	Handling Concept Drift	Online/Offline	Yes	Offline	Ensemble Classifier	Classification
Suarez-Tangil et al. (2015)	Both	SL	ONB, J48, K-means	Offload or not to Offload	Online	Yes	Both	Naive Bayes, J − 48, K-means	Classification/Clustering
Boukhechba et al. (2015)	Online	SL	Habit Tree Data Structure	To Optimize and Predict Next Location	Online	Yes	Online	Association Rule Mining (ARM)	ARM
Haghighi et al. (2013)	Online	SL/UL	Multiple Learning Algorithms	On-board Data Stream Mining	Online	Yes	Online	ARM, Classification, Clustering	Light-weight Data Mining
Gomes et al. (2012b)	Online	SL	Naive Bayes	Model Personalization	Online	Yes	Online	Naive Bayes	Classification
Liu et al. (2012)	NA	SL/UL	Multiple Learning Algorithms	Proof of concept for Mobile Data Mining	Online	Yes	NA	Mobile WEKA library	Classification, Clustering, ARM
Khan et al. (2010)	Offline	SL	Feed Forward Neural Network	On-board Multi-sensor Activity Recognition	Online	Yes	Offline	Feed Forward Selection	Classification
Mukherji et al. (2014)	Online	SL	Tree based	To Perform Context Prediction	Online	Yes	Online	Sequential Pattern Mining	Sequential Pattern Mining
Abdallah et al. (2012)	Both	SL/UL	Incremental/Active Learning	Clustering	Online	Yes	Offline	K-means	Clustering
Sidek et al. (2014)	Both	SL	NB, BN, and MLP	ECG Signal Classification for Biometric Identification	Online	Yes	Online	BN, MNN	Classification
Khan et al. (2013)	Offline	SL	Feed Forward Neural Network	On-board Multi-sensor Activity Recognition	Online	Yes	Offline	Feed Forward Selection Algorithm	Classification
Srinivasan et al (2014)	Online	SL	Tree based	To Perform Context Prediction	Online	Yes	Online	Sequential Pattern Mining	Sequential Pattern Mining
Siirtola and Roning (2013)	Offline	SL	Decision Tree and QDA	Model Training	Online	Yes	Offline	Decision Tree and QDA	Classification
Yang et al. (2014)	Offline	SL	SVM	Model Training	Online	Yes	Offline	Support Vector Machine	Classification
Lu et al (2012)	Offline	SL	Gausian Mixture Model	Speaker Identification	Online	Yes	Offline	GMM, K-means, and EM	Classification
Donohoo et al. (2014)	Offline	SL	LDA, LLR, SVM, VRL	Model Training	Online	Yes	Offline	LDA, LLR, SVM, KNN	Classification
Oshin et al. (2015)	Offline	SL	EHMS	Model Training	Online	Yes	Online	EHMS	Classification
Rai et al. (2012)	Online	SL/ UL	K-means and SVM	Higher Order Feature Extraction/ Model Training	Online	Yes	Online	Support Vector Machine	Classification
Wang et al. (2012)	Offline	UL	Emerging Patterns	Activity Recognition	Offline	Yes	Offline	Emerging Patterns	Classification
Gaber et al. (2014b)	Offline	SL	Hoefding Tree	Distributed Classification	Online	Yes	Offline	Emerging Patterns	Classification
Ortiz et al. (2015)	Online	SL	NA	NA	Online	No	NA	K-means	Distributed Clustering
Braojos et al. (2014)	Offline	SL	NFC	Classification	Online	No	Offline	Nero Fuzzy Classifier, DT	Multi-level Classification

Table 4 (*Continued*)

Reference	Learning Mode	Learning Type	Learning Algorithm	Learning Objective	Mining Mode	Learning Model	Training Mode	Data Mining Algorithm	Purpose
Min and Cho (2011)	Offline	SL	SVM, NB, DT	Classification	Online	Yes	Offline	SVM, NB, DT	Multi-level Classification
Stahl et al. (2012)	Offline	SL	Hoefding Tree	Distributed Classification	Online	Yes	Offline	Hoefding Tree	Distributed Classification
Jayaraman et al. (2014b)	NA	NA	NA	NA	NA	NA	NA	Light-weight Algorithms	Classification, Clustering, ARM
Dou et al. (2011)	NA	NA	NA	NA	Online	No	NA	K-means	Distributed Clustering
Wu et al. (2013)	Offline	SL	EM	Classification	Offline	Yes	Offline	HMM	Classification
Eom et al. (2015)	Online	SL	Instance based Learning, Naive Bayes, Single Layer Perceptron	Classification	Online	Yes	Online	Instance based Learning, Naive Bayes, Single Layer Perceptron	Machine Learning based Dynamic Task Scheduling
Sherchan et al. (2012)	Online	UL	Light-weight Clustering	Change Detection	Online	Yes	Online	Light-weight Algorithms	Light-weight Data Mining
Jayaraman et al. (2014a)	Online	SL/UL	Light-weight Clustering and Classification	Local Analytics	Online	Yes	NA	Light-weight Algorithms	Light-weight Algorithms
Lin et al. (2013)	NA	NA	NA	NA	online	No	NA	K-means	Clustering of GPS Data
Yuan and Herbert (2014)	Both	SL/UL	NB, DT, Nearest Neighbor, Neural Network	Universal and Personalized Model Development	Online/Offline	Yes	Both	NB, DT, Nearest Neighbor, Neural Network	Personalized Activity Classification
Hassan et al. (2015)	Online	SL	MLP, Linear Regression, SVM, Decision tree	Model Training for Predicting Offload-able Computations	Online	Yes	Online	MLP, LR, SVM, DT	Compute/Resource-intensive Methods Classification
Talia and Trunfio (2010)	Offline	NA	Numerous	NA	Offline	NA	Offline	Numerous	Multiple
Kargupta et al. (2010)	NA	NA	NA	NA	Online	No	NA	Correlation and Distance Matrices Computations	Change Detection
Yoon (2013)	NA	NA	NA	NA	Online	No	Online	Multi-level Deployment of ARM Algorithms	ARM
Gu et al. (2011)	Offline	UL	Emerging Patterns	Prediction of Emerging Patterns	Online	Yes	Offline	Emerging Patterns	ARM

Table 5
Knowledge management heterogeneity.

Reference	Local DM	Remote DM	On-screen Visualization	Remote Visualization	Knowledge Management
Pasricha et al. (2015)	Y	N	Y	N	Online
Abdallah et al. (2015)	Y	N	Y	N	Online
Boukhechba et al. (2015)	Y	N	N	N	N
Haghighi et al. (2013)	Y	Y	N	N	Online
Gomes et al. (2012b)	Y	N	Y	N	Online
Mukherji et al. (2014)	Y	N	N	N	N
Abdallah et al. (2012)	Y	N	N	N	N
Srinivasan et al (2014)	Y	N	N	N	N
Yang et al. (2014)	Y	N	Y	N	Online
Gaber et al. (2014b)	Y	N	Y	N	Online
Min and Cho (2011)	Y	N	N	N	N
Stahl et al. (2012)	Y	N	Y	N	Online
Jayaraman et al. (2014b)	Y	Y	N	Y	Both
Dou et al. (2011)	N	N	Y	N	N
Wu et al. (2013)	N	Y	N	Y	Offline
Sherchan et al. (2012)	Y	Y	Y	Y	Offline
Jayaraman et al. (2014a)	Y	Y	Y	Y	Offline
Lin et al. (2013)	N	N	N	NA	Online
Yuan and Herbert (2014)	Y	N	N	NA	Both
Talia and Trunfio (2010)	N	Y	Y	N	Offline
Kargupta et al. (2010)	Y	Y	Y	Y	Online
Yoon (2013)	Y	Y	NA	NA	Online

based adaptive strategies work by monitoring the velocity of incoming and outgoing data streams. These adaptive data stream mining algorithms adjust the execution behavior according to data rates. The memory and CPU based adaptation strategies work by profiling the computational requirements of data stream mining algorithms and adjusting the execution behavior accordingly. The context-aware adaptive strategies models different situations and adjust the execution behavior of data stream mining systems when a relevant situation is inferred. The learning model based strategies consider the execution history of data stream mining applications, learn the execution patterns, and alter the execution behavior according to predicted settings. Event based strategies work by adopting the execution behavior of data stream mining algorithms accordingly when a specific event occurs. Table 6 presents a detailed literature review of adaptation strategies in MDSM platforms.

6.2. Application partitioning

Distributed MDSM applications are partitioned to run on multiple devices and systems. The application partitioning strategies are controlled by either far-edge devices, cloud servers, or edge servers. MDSM applications are either partitioned dynamically at runtime after assessing the resource requirements of the running processes or the application is partitioned in fixed form where specific application components run at designated devices and systems.

The applications are partitioned either on the basis of data or computations. The data based application partitioning is performed by executing data parallel strategies where partial data streams are offloaded and executed in various devices and systems. The computation based partitioning is performed by measuring the computational requirement and granularity of data stream mining algorithms. In computation based partitioning partial tasks such as methods, classes, programs, and applications are executed in various device and systems. Application partitioning is performed either offline or online. The offline partitioning is performed before or after the application execution however the online partitioning is performed during the application execution process. Table 7 presents the detailed literature review of application partitioning methods in MDSM platforms.

6.3. Computation offloading

Existing computation offloading schemes are based on different communication models that vary in terms of client-server settings, virtual machine migration, and mobile agent configurations (Khan, 2015). In client-server based settings, offloading components reside on the

Table 6
Adaptation heterogeneity.

Reference	System Level	Algorithm Level	Data Rate	CPU	Memory	Context	Learning Model	Event
Pasricha et al. (2015)	Y	N	N	Y	N	N	Y	Y
Abdallah et al. (2015)	N	Y	N	N	N	N	Y	N
Boukhechba et al. (2015)	N	N	N	N	N	N	Y	N
Haghighi et al. (2013)	Y	Y	Y	Y	Y	Y	N	N
Gomes et al. (2012b)	N	Y	N	N	N	N	Y	N
Lu et al (2012)	N	N	N	N	N	N	Y	N
Gaber et al. (2014b)	N	Y	Y	Y	Y	N	N	N
Stahl et al. (2012)	N	Y	Y	Y	Y	N	N	N
Jayaraman et al. (2014b)	Y	Y	Y	Y	Y	N	N	N
Eom et al. (2015)	N	N	N	N	N	N	Y	N
Sherchan et al. (2012)	Y	Y	Y	Y	Y	Y	Y	N
Jayaraman et al. (2014a)	Y	Y	Y	Y	Y	Y	Y	N
Yuan and Herbert (2014)	Y	N	N	N	N	N	Y	N
Kargupta et al. (2010)	N	Y	N	N	N	N	N	N

Table 7
Application partitioning heterogeneity.

Reference	Device	Cloud	Server	Edge	Type	Partitioning Mode	Model	Granularity	Form
Wang et al. (2012)	Y	NA	NA	NA	Offline	Data-based	Static	Data	Fixed
Gaber et al. (2014b)	Y	N	N	N	Offline	Data-based	Static	Data	Fixed
Ortiz et al. (2015)	Y	N	NA	NA	Offline	Data-based	Static	Data	Fixed
Braojos et al. (2014)	Y	N	N	N	Offline	Computation-based	Static	Learning Model	Fixed
Min and Cho (2011)	Y	N	N	N	Offline	Data-based	Static	Data	Fixed
Stahl et al. (2012)	Y	N	N	N	Offline	Data-based	Static	Data	Fixed
Jayaraman et al. (2014b)	Y	Y	Y	Y	Offline	Data-based	Dynamic	Data	Fixed
Dou et al. (2011)	Y	N	N	N	NA	Data-based	Static	Data	Fixed
Sherchan et al. (2012)	Y	Y	N	N	Offline	Computation-based	Static	Application	Fixed
Jayaraman et al. (2014a)	Y	Y	N	N	Offline	Computation-based	Static	Application	Fixed
Lin et al. (2013)	N	N	Y	N	Offline	Data-based	Static	Data	Fixed
Yuan and Herbert (2014)	Y	Y	Y	NA	Offline	Data-based	Static	Data	Fixed
Hassan et al. (2015)	Y	No	No	No	Online	Method-based	Dynamic	Method	Dynamic
Talia and Trunfio (2010)	N	N	Y	N	Offline	Data-based	N	N	N
Yoon (2013)	Y	Y	NA	NA	Offline	Data-based	Static	Method	NA

mobile device that offloads the computations after performing collaborative cost-benefit analysis for computation offloading favorability. Cost-benefit analysis is performed to label the local and remote computations for application partitioning (Liu et al., 2013) and resource-hungry computational tasks are offloaded to the nearest or designated surrogates (servers) in the cloud. The main concern with server-based computation offloading is the requirement for pre-installed cloud services in ad-hoc cloud environments. In virtual machine migration-based communication models, the memory image of a central cloud server is migrated in cloudlets, which lowers the communication cost as well as overall bandwidth utilization in highly-dense mobile cloud computing environments (Satyanarayanan et al., 2009). However, live virtual machine migration introduces latency in service provisioning (Ahmad et al., 2015). In addition, the preservation and resumption of application states during migration is also a major challenge (Satyanarayanan et al., 2009). In mobile agent communication models, the application clones are migrated in cloud environments to augment the mobile devices with cloud resources. However, mobile agent management and clone security are the main issues in mobile agent-based mobile cloud computing environments (Khan, 2015).

Computation offloading schemes function with single-site and multiple-site surrogate settings (Abolfazli et al., 2014). In the case of single-site surrogates, the application components are offloaded to the same server in the mobile cloud computing architecture. However, this setting develops a tightly bounded relationship between mobile applications and their corresponding surrogates. Therefore, the dynamic mobility increases the latency in distant mobile devices (Abolfazli et al., 2014). On the other hand, multiple-site surrogates work in two ways. Either application clones are provided at multiple sites using live virtual machine migration methods or different program components are executed at various surrogates. In addition, the virtual machine migration problem also brings the parallelization challenge, which needs to be addressed in multiple-site surrogates (Abolfazli et al., 2014). The programs should be effectively partitioned and mapped into graph data structures that are further optimized for seamless application execution in mobile cloud computing environments. In addition, adaptive computation offloading schemes consider program execution contexts and previous program instances and devise optimal execution strategies accordingly. Adaptive computation offloading schemes consider various parameters, including network connections and bandwidths, workloads, architectural heterogeneity and task deadlines. However, the favorable offloading decision becomes complex due to varying bandwidth, resource availability and network dynamics (Abolfazli et al., 2014).

Computation offloading schemes in mobile cloud computing environments are categorized as either static or dynamic (Kumar et al., 2013). In static schemes, one-time cost-benefit analysis is performed and offloading-favorable computations are offloaded in mobile cloud computing environments. Dynamic computation offloading schemes initially perform a cost-benefit analysis, implement online profiling, tag the offloadable program components during application execution, perform application partitioning for local and remote execution, and offload the computation offloading favorable components in mobile cloud computing environments. Computation offloading is performed at different granularity levels. At the coarse-grained level, entire applications are offloaded in mobile cloud computing environments. The coarse-grained level computation offloading is well-suited when cloud resources are available at one-hop distances from mobile devices. However, in the case of cloudlets, live virtual machine migration may incur higher cost in terms of latency. On the other hand, the complete migration of entire application states in edge servers increases local computation costs. At fine-grained levels, computation offloading is performed at various application code levels, including method, task, object, thread, class and program levels. These different granularity levels increase the decision complexity of computation offloading. Optimal computation offloading strategies involve multiple offloading objectives, including performance enhancement, energy gain, reduced execution time, minimum bandwidth utilization cost, and data reduction, among others. Table 8 presents the detailed literature review of existing computation offloading methodologies for MDSM platforms.

6.4. Data transfer

MDSM applications transfer data streams among devices and systems in multiple ways. The simplest data stream transfer strategies are based on transferring raw data streams. The raw data streams are either stored on-board or directly collected from data sources. Sometimes the MDSM applications perform initial data processing and transfer the intermediate data to other systems and sometimes the data stream mining algorithms are executed onboard in light-weight processing modes

Table 8
Computation offloading heterogeneity.

Ref.	Mode	Type	Parameters	Offloading Devices	Servers	Objective
Wang et al. (2012)	Offline	Static	NA	Single	Single	Pattern Mining
Gaber et al. (2014b)	Offline	Static	NA	Single	Multiple	Collaborative Mining
Ortiz et al. (2015)	Online	Static	Execution Time	Multiple	Multiple	Collaborative Clustering
Stahl et al. (2012)	Offline	Static	NA	Single	Multiple	Collaborative Mining
Jayaraman et al. (2014b)	Both	Dynamic	Connectivity	Multiple	Multiple	Multi-objective
Eom et al. (2015)	Online	Dynamic	Classifier	Single	Multiple	Dynamic Scheduling
Sherchan et al. (2012)	Offline	Static	NA	Multiple	Single	Reduced Data Collection
Jayaraman et al. (2014a)	Offline	Static	NA	Multiple	Single	Reduced Data Collection
Hassan et al. (2015)	Online	Dynamic	Multiple	Single	Multiple	Reduced Latency and Energy Gain
Talia and Trunfio (2010)	N	N	NA	Multiple	Multiple	Offline Remote Data Analysis
Yoon (2013)	Offline	Static	NA	Multiple	Single	Multi-layer Data Mining

and resultant knowledge patterns are transferred to other devices and systems for aggregation and global knowledge view.

The data streams are transferred in push-based, pull-based, on-demand, or opportunistic settings. In push based strategies, the mobile devices simply transfer the data stream to connected devices and systems. In pull based strategies, remote systems like cloud servers monitor the connections and periodically collect the data stream from mobile devices. The on-demand strategies work when the remote servers issue a query for data processing or sensing to connected mobile devices which in turn perform the required operations and communicate the results back to requesting server. On-demand data transfer strategies are useful for mobile crowd sensing applications. The opportunistic data transfer strategies monitor the connected devices and systems and find the feasible environment for pushing or pulling data streams among connected devices and systems. Smart data reduction is another approach for data transfer where mobile devices perform the data stream mining operations and the results are communicated only if there is a significant change in the data stream. Table 9 presents the summary of data transfer strategies in MDSM platforms.

In this section, we presented a thorough literature review of methods that are used to handle the heterogeneity in MDSM platforms.

7. Literature summary

This section presents the summarized view of major contributions relevant to MDSM platforms. Table 10 presents the comparison of these contributions.

7.1. MineFleet

Minefleet is a distributed data stream mining platform for vehicular data stream analysis (Kargupta et al., 2010). The data stream mining components reside in an onboard computing system that continuously mine the data streams which is acquired from on-board vehicle sensors. Minefleet is based on five components: (a) Onboard hardware component enables data acquisition form multiple onboard sensors and provides the communication interface for data transfer, (b) onboard data stream mining and management module enables the execution of various data stream mining and statistical data analysis algorithms and in case of unusual behaviors in vehicle data the module enables to connect with remote MineFleet servers located in centralized data center, (c) MineFleet server collects the analytics results from vehicles to perform further analysis, (d) MineFleet web services provide application programming interface (API) to access and view the analyzed data from MineFleet servers and (e) privacy module manages the end-to-end privacy in MineFleet system.

Minefleet uses light-weight algorithms to handle online data streams and performs in-memory computations for finding the distance metrics, invariance, correlation, and inner product between data stream elements. In addition, MineFleet performs change detection using correlation matrix to uncover the unusual behavior from vehicular data streams. Overall, MineFleet is designed to reduce the onboard data storage cost and data communication cost in wireless networks. In addition, the system processes the high-volume data streams on resource constrained onboard computing system. However, it is purposefully built for vehicular data stream mining applications and still lacks the generality for execution of heterogeneous data stream mining applications.

7.2. OMM

Open Mobile Miner (OMM) is a situation aware and adaptive data stream mining system for mobile devices (Haghighi et al., 2013). OMM architecture enables six main components: (a) data source component generates data stream from four different sources which include onboard sensors, controlled data streams generated by OMM applications, recording and replaying data stream using CSV files, and web services, (b) data stream capture component acquires data stream from different

Table 9
Data transfer heterogeneity.

Reference	Push-based	Pull-based	On-demand	Opportunistic	Smart Data Reduction
Wang et al. (2012)	Yes	N	N	N	N
Gaber et al. (2014b)	Yes	N	N	N	N
Ortiz et al. (2015)	Yes	N	N	N	N
Stahl et al. (2012)	Yes	N	N	N	N
Jayaraman et al. (2014b)	Yes	Yes	Yes	N	N
Eom et al. (2015)	Yes	N	N	N	N
Sherchan et al. (2012)	Yes	N	N	N	N
Jayaraman et al. (2014a)	Yes	N	Yes	N	Yes
Lin et al. (2013)	Yes	N	N	N	N
Hassan et al. (2015)	Yes	N	N	Yes	N
Talia and Trunfio (2010)	Yes	N	N	N	N
Yoon (2013)	Yes	N	N	N	N

Table 10
Strengths and weaknesses of existing MDSM platforms.

Platform	Model	Strengths	Weaknesses
MineFleet	MCC	- Distributed	- Difficult to generalize
		- Onboard data mining	- Supports only vehicular onboard applications
		- Data reduction at mobile end	- Highly coupled applications
		- Reduced bandwidth utilization	- Dependency over Internet connections
OMM	Far-edge	- Mobile-based	- Compromises on knowledge quality
		- Adaptive data processing	- Does not supports heavy-weight data processing
		- Light-weight algorithms	- Privacy concerns
		- Component-based architecture	
CARDAP	MCC	- Distributed	- Does not performs runtime load-balancing
		- Onboard data processing	
		- Data reduction	
		- Light-weight data mining algorithms	
MOSDEN	MCC	- Distributed	- Lacks general heavy-weight components
		- Data filtration	- Only focuses on data acquisition and data processing heterogeneity
		- On-demand access to data	
MARS	Far-edge	- Mobile-based	- Lacks generality
		- Adaptive	- Does not handles heterogeneity
		- Specific to activity recognition applications	
STAR	Far-edge	- Mobile-based	- Lacks generality
		- Adaptive	- Does not supports heavy-weight data processing
		- Specific to activity recognition	- Does not handles heterogeneity
PDM	F2F	- Mobile-based	- Lacks heavy-weight data processing
		- Distributed	- Does not uses cloud services
		- Agent-oriented	- Does not handles heterogeneity at application level
		- Light-weight data processing	
		- Scalable	
CARA	MCC	- Distributed	- Lacks in generality
		- Context-aware data analysis	- Lacks load-balancing
		- Specific to activity recognition	- Does not handles heterogeneity at platform level
		- Provides universal learning models	
SOA	MCC	- Cloud-based data analytics services	- Does not handles heterogeneity in MDSM applications
MobiSens	MCC	- Distributed	- Does not handles heterogeneity at platform level
		- Generic sensing architecture	
		- Specific to activity recognition applications	
		- Thin Clients	
		- Two-tier back-end architecture	
MSM	Far-edge	- Mobile-based	- Lacks support from cloud servers and other devices
		- Enables general components for association rule mining	
Mobile Miner	Far-edge	- Mobile-based	- Lacks support from cloud servers and other devices
		- Supports wearable and IoT devices	
		- Mines co-occurrence patterns	
Three-tier	Hierarchical	- Multi-layer architecture	- High Coupling
		- Offers data mining services at multiple levels	- Lacks in handling heterogeneity

data sources and input either directly into light-weight data mining algorithms or redirects through adaptation engine if OMM is operating in adaptive mode, (c) resource monitor component tracks the memory, CPU, and battery power in mobile devices for seamless execution and adaptations, (d) a library of light-weight data stream mining algorithms, (e) a library to enable visualization facilities in mobile devices, and (f) adaptation engine to execute resource and situation aware adaptation strategies. OMM provides light-weight classification, clustering, and association rule mining algorithms however the adaptation strategies are occasionally required to handle the resource constraints in mobile devices. Situation-aware adaptation strategies maintain a set of predefined situations which are inferred from periodically collected contextual information. Alternately, resource-aware adaptation strategies control the data rate and execution behaviors of data stream mining algorithms on the basis of incoming/outgoing data rates, and memory and CPU availability. OMM is a first general purpose mobile based adaptive data stream mining system but the light-weight execution of data mining algorithms enforces the compromises over the quality of knowledge patterns such as level of accuracy of classifiers, the number of clusters produced by clustering algorithms or association rules found by association rule mining algorithms.

7.3. CARDAP

Context aware real time data analytics platform (CARDAP) is a distributed data stream mining system for mobile crowd sensing applications (Jayaraman et al., 2014a). CARDAP architecture offers five key

components: (a) data stream capture component facilitate in data acquisition from range of on-board and off-board, physical and virtual, and sensory and non-sensory data sources, (b) analytic component facilitates in deploying application specific data stream mining algorithms such as activity recognition components, (c) open mobile miner to facilitate generic execution of light-weight data stream mining algorithms, (d) data sink component to push data to external data stores such as cloud based data storage services, and (e) storage and query component to store results of local analytics for lateral on-demand querying. CARDAP uses three data transfer strategies from mobile devices to cloud data stores. First, it uses naive approach to uploading raw data stream in the cloud environments. Second, it uses local analytics approach where locally stored analytics results can be acquired by cloud data stores using on-demand querying. The Third approach enables smart data reduction where analytics results are transferred in case of significant change.

7.4. MOSDEN

Mobile sensor data engine (MOSDEN) is a component based platform to facilitate opportunistic sensing applications in mobile crowd sensing environments (Jayaraman et al., 2014b). MOSDEN architecture is based on seven components: (a) plugins are the software applications that independently facilitate in interfacing with different data sources, (b) virtual sensors represent the abstraction layer of physical data sources, (c) processors components facilitate in development of learning models and mining algorithms, (d) storage components facilitate in data storage from virtual sensors and processors components, (e) query manager component answers and resolves queries from external sources, (f) service manager establishes the persistent and non-persistent data transfer strategies from MOSDEN clients to external data sources and (g) API components provide application programming interfaces to external applications for data access from MOSDEN clients. The component based design of MOSDEN offers generality and programmability of the proposed architecture. In addition, local data processing and interaction with plethora of physical and virtual data sources helps in the deploying a wide range of mobile opportunistic sensing applications.

7.5. MARS

Mobile activity recognition system (MARS) process the data stream acquired from onboard mobile phone accelerometer (Gomes et al., 2012b). MARS extracts statistical features and labels the data stream with certain physical activities such as walking, running, and standing. MARS builds learning model from the training data and uses it for future activity recognition. The system facilitates in model personalization using dynamic model adaptation for personal data streams. The incremental learning method is adapted and models are regenerated to accommodate changes in the data stream.

7.6. Star

Stream learning for mobile activity recognition framework, named as *star*, addresses the issue of change detection from streaming data for accurate activity classification (Abdallah et al., 2015). The framework uses incremental learning approach to handling concept drift over evolving streaming data. The *star* framework offers three components to build learning models, perform activity recognition, and to adapt with evolving data streams. First, the modeling component builds the initial sets of clusters using supervised learning approach. Later-on fine-grained clusters are obtained using unsupervised learning models. Second, the recognition components perform activity recognition over sliding windows data using modeling components. Third, adaptation component performs active online learning if the recognition component outputs a new activity. The *star* handles concept drift effectively however its functionality is limited to mobile activity recognition applications.

7.7. PDM

Pocket data mining (PDM) is an agent-oriented distributed data stream mining system for mobile devices (Gaber et al., 2014b). PDM architecture is based on three generic software agents. First, mobile agent miners (AM) are used to implement the data stream mining algorithms in mobile devices. The AMs also facilitate in batch learning models to handle the historical data. Second, mobile agent resource discoverers (MRD) perform resource and task discovery operations throughout the ad-hoc network. MRD matches the data sources with stream mining algorithms controlled by AMs and it makes a decision about which AMs should run in given mobile devices. Third, mobile agent decision makers (MADM) travel through the ad-hoc network, collect the results of processed data streams, and perform on-the-fly knowledge integration. PDM applications work in following steps. First, a mobile device initiates a data mining task by establishing an ad-hoc network and activating the MRDs. The MRD discovers resources, matches the data sources, decides which AMs should be executed on connected devices, and runs the data mining tasks. Finally, the MADM travels through ad-hoc network and integrates the acquired results. PDM offers seamless distributed data stream mining, however, continuous executions of mobile agents increase energy and resource consumption in mobile devices.

7.8. CARA

Context aware real-time assistant (CARA) provides a cloud based data analytics tool for mobile activity recognition applications (Yuan and Herbert, 2014). CARA works by downloading a global learning model for activity recognition from cloud to mobile device. The mobile device collects raw data stream, performs segmentation with 50% overlapping sliding windows and computes features from each sliding window. CARA distinguishes the static and dynamic activities using threshold-based method wherein threshold is defined on the intensity of accelerometer signals. CARA recognizes static activities through threshold based method however it implies classification algorithms such as decision trees, Bayesian networks, nearest neighbors, and neural networks to predict dynamic activities. CARA components run on both mobile devices and cloud servers. The mobile devices download the universal model and perform classification operations. The new unclassified data streams are stored temporarily in local storage and uploaded to cloud servers when a Wi-Fi connection is available. The cloud servers provide blob storage, enables queuing mechanism, and utilize multiple nodes to produce best classification models for each user.

CARA implements five types of queues: (a) data queue for client-controller communication, (b) result queue to select suitable model for each individual user, (c) register queue to register new users, (d) task queue to control the task execution in universal node, and (e) model queue to retrain the classification models and update the relevant information. CARA designates four types of nodes in the cloud environments: (a) controller node controls the flow of incoming data, (b) machine learning node facilitates the model training process, (c) universal node aggregates the data from all users and builds a universal classifier and (d) evaluation node selects the most suitable classifier for individual user.

7.9. SOA

Service oriented architecture (SOA) is based on client-server communication model where mobile devices request for specific data mining services and cloud servers provide relevant web services (Talia and Trunfio, 2010). The proposed SOA architecture provides three types of components: (a) data providers, (b) mobile clients, and (c) mining servers. The data providers are the applications that generate the data streams and mobile clients are the requesting applications that require some specific mining tasks to be performed at mining servers.

7.10. MobiSens

MobiSens provides a generic sensing architecture for large-scale activity recognition (Wu et al., 2013). MobiSens architecture is based on client server communication model where mobile devices work as clients and back-end server architecture works at two tiers. Major components at mobile client include data widgets to sense raw data streams, data aggregators to buffer and store data streams until the availability of Internet connections, and sensing profile pulling component that pulls the sensing profiles of MobiSens applications to configure list of sensors, sampling data rates, strategy for data sampling, interval between data push operations and many others. On the server side, MobiSens components perform data storage, indexing, and heavy weight data processing at the first tier. On the second tier, MobiSens server facilitates in remote applications such as life logging, community sensing etc.

7.11. Mobile WEKA

Mobile WEKA is a general purpose tool developed to show the mobile implementation of WEKA data mining library (Liu et al., 2012). The mobile application facilitates in performing classification, clustering, and association rule mining operations. Mobile WEKA shows the proof of concept for mobile devices as data mining platforms, however, it lacks the generality and addressing heterogeneity at the application level.

7.12. MSM

Mobile Sequence Miner (MSM) provides a general purpose tool for mining association rules from frequent activity sequences (Mukherji et al., 2014). MSM collects the data stream for application usage, location, and call logs to infer the context and sequence of activities. MSM runs as a back-end service in android mobile phones to continuously monitor context and find frequent sequences. MSM application execute in three steps: (a) preprocessing operations on incoming data streams are performed to find the interleaved context items, (b) sequence databases of interleaved context events is generated, and (c) frequent sequences are generated from sequence databases.

7.13. MobileMiner

MobileMiner facilitates in mining co-occurrence patterns from GPS locations, call logs, and application logs to infer the contextual information (Srinivasan et al., 2014). MobileMiner runs as a back-end service in Tizen applications which run in multiple platforms such as wearable devices, home appliances, and mobile IoTs. MobileMiner extracts the time-stamped baskets (chunks of the data stream) using the base basket extractor component whereby co-occurring contexts are stored in each basket. MobileMiner uses base rule miner component which mines stream of extracted baskets and uncovers underlying co-occurrence patterns. MobileMiner enables some other components such as app usage filter and app rule miner to retrieve application usage relevant context baskets and find co-occurring patterns. MobileMiner communicates with external devices and systems using pattern retriever component that retrieves overall and detailed patterns. The prediction engine component in MobileMiner retrieves prediction information from overall patterns.

7.14. Three-tier data mining architecture

Researchers proposed MDSM architecture that works at three layers (Yoon, 2013). The small-scale micro-controller devices at the lowest layer enable the sensing operations, perform row-level instance based learning, and execute data filtration methods. The filtered data streams are transferred to user smartphones which find the local patterns using onboard computational resources. In addition, the smartphones correlate local patterns to form regional patterns. The cloud servers at the highest layer integrate the regional patterns from multiple smartphones to generate the global patterns. The proposed layered architecture is suitable for many application area such as patient health monitoring systems, community sensing, and mobile crowd sensing applications.

In this section, we presented the summary literature of few recently proposed MDSM platforms. In the next section, we present the detailed gap analysis of existing research work in order to articulate the future research directions for data stream mining in mobile edge cloud computing systems.

8. Gap analysis and future research directions

The heterogeneity needs to be controlled at both application and platform level.

8.1. Controlling heterogeneity at application level

Ideally, MDSM applications should collect and process data stream using on-board memory to minimize the efforts in data storage and reduce the latency which occurs due to I/O operations. However, complexities introduced by data acquisition strategies and resource constraints in mobile devices are the main bottlenecks in performing in-memory application execution. For high volume and high speed data streams, the chunking and segmentation operations are performed using fixed size sliding windowing methods. The settlement of windows size is challenging because of varying complexities and operational behaviors (e.g. nature of data structures such as arrays, trees, graphs, data storage in random, sorted, unsorted, compressed arrangements, traversal behaviors such as search strategies) of data preprocessing and mining algorithms. In addition, fixed size windows may create latency when data streams enter with variable data rates. In addition, MDSM applications need to handle the heterogeneity for the acquisition of data stream from authentic data sources, in known data formats, however, the existing literature lack to address these issues.

The data fusion strategies increase/decrease the lateral complexities in stream execution process. Existing literature mainly presents early data fusion strategies which increase the complexity in the system. The raw data collection from multiple data source induces noise, outliers, and missing values which increase the level of sparsity in high-dimensional data streams. Therefore, dimension reduction, anomalies and outliers detection, and sketching operations are used to preprocess and improve the quality of data streams. Late data fusion helps in reducing the complexities by performing initial data preprocessing and fusing reduced, noise free, and complete data points. Discriminatory data fusion strategies help in improving the performance of the system by selecting useful attributes of the data stream that help in uncovering quality knowledge patterns. However discriminatory data fusion may result in compromises over the quality of knowledge patterns such as accuracy

of classifiers, the number of itemsets produced by frequent pattern mining algorithms, and the number of clusters in data clustering algorithms. Therefore more research is needed for late and discriminatory data fusion by keeping a balance between quality of knowledge patterns and resource consumption of MDSM applications.

Although data preprocessing operations vary according to application requirements and objectives of data mining algorithms however existing literature mainly considered feature extraction and noise filtration as preprocessing methods. Existing literature still lacks in preprocessing methods for dimensionality reduction, outliers, and anomalies detection methods. The dimension reduction methods could help in reducing the computational complexities of MDSM applications by projecting high dimensional complex data streams in low dimensional feature vectors. In addition, dimension reduction methods help in reducing the sparsity which increase the computational complexity in MDSM applications. Highly sparse data streams need to construct large learning models and consume more computational resources, therefore, effective reduction of high dimensional data reduce the resource consumption.

MDSM applications should perform all data mining operations in-memory. However, most of the existing studies first preprocess and store the data stream before training the learning models. Few studies performed in-memory training in batch mode or updated model from generalized to personalized models, however, existing literature still lacks in the online (re-)training of learning models in mobile devices. Aside from this, the prediction by data stream mining algorithms is made online by keeping the learning models in memory, evaluating the attributes of incoming data streams and uncovering new knowledge patterns. However existing studies used same data stream for training and prediction, therefore, lacks in generality to adapt new data streams.

Existing studies lack in knowledge management features and none of the studies focuses knowledge management as a core issue. However, there exists a need to handle knowledge management issue effectively. Distributed application logic in MDSM platforms generate knowledge patterns in different computing environments, therefore, integration and summarization of relevant knowledge patterns requires further attention.

8.1.1. *Critical factors of complexity in MDSM applications*

Numerous factors affect the complexity of MDSM applications. Since the mobile applications execute in resource constrained environments therefore high volume of incoming data stream becomes a critical factor to handle the complexity in mobile devices. Although existing methods use light-weight algorithms which do not consider the whole data stream and reduce the quality of knowledge patterns. However high data size severely impacts the heavy weight MDSM algorithms. Likewise high data rate in MDSM applications increases the computational complexity. Exiting MDSM applications work online by performing in-memory operations with time constraints. The algorithms are executed as one-pass algorithms with the condition that current data stream must be processed before the arrival of next data stream.

The choice of data fusion strategy helps in increasing or decreasing the computational complexity of MDSM applications. Early data fusion strategies produce redundant, noisy, and anomalous data streams because MDSM applications collect data streams without any filtering and/or preprocessing methods. On the other hand, late and discriminatory data fusion strategies produce high quality data streams hence requires fewer computations at later stages.

The operational behaviors such as populating data structures, the traversal methods, and nature of computational operations affect the computational complexity of data preprocessing and data mining algorithms. The computational complexity increases when the data stream mining algorithms are bounded to perform all operations using on-board computational resources. Existing systems use light-weight algorithms, that use shallow data structures and linear traversal behaviors, to handle the computational complexity of data preprocessing and mining algorithms.

The complexity of MDSM applications also increases during the learning phase. Online learning over large streaming data becomes computationally infeasible due to resource limitations and constraint of keeping whole data stream in memory. The behavior of learning model such as supervised, unsupervised, and semi-supervised settings also affect the computational complexity of the MDSM applications. The supervised and semi-supervised learning model initially uses labeled data stream hence learning algorithms are trained with-in a confined feature space. On the other hand, during unsupervised learning the leaning models need to be trained with high-dimensional complex data streams which quickly hamper the computational resources especially in mobile devices.

The high complexity in aforementioned critical factors impacts the MDSM applications as a whole as shown in Fig. 10. The large size of data stream impacts the complexity of data rates, preprocessing algorithms, learning behaviors, and data mining operations. Likewise, the increase in computational complexity at any stage impacts the subsequent operations in MDSM applications.

8.2. *Controlling complexity at platform level*

Ideally, MDSM applications should perform maximum computational operations near the data sources without latency. However, the resource limitations in mobile devices enforce to acquire computational support from other mobile devices and large scale computing infrastructures such as clouds, grids, and Internet enabled servers. Existing systems for MDSM works adaptively in mobile environments. Alternately, the systems enable distributed data stream mining in mobile cloud settings. However existing literature still lacks the systems which fully utilize the capabilities of far-edge mobile devices, edge servers, and cloud computing architectures. In addition, the execution models of existing systems are designed as standalone, distributed, collaborative, and parallel settings. However existing literature still lacks the device-centric systems based on collaborative and distributed execution model for MDSM applications.

Due to variations in computational complexities of MDSM applications and required quality of knowledge patterns, MDSM applications enable light-weight processing for efficient resource utilization in mobile devices. Alternately, the algorithms are made adaptive by adjusting the processing behaviors according to incoming data rates, required quality of knowledge patterns, resource availability, and outgoing data rates. Although algorithm level adaptations efficiently handle the resource limitations, however, customization of each algorithm impacts generality. System level adaptation strategies can help in achieving the generality however existing literature still lacks relevant methods.

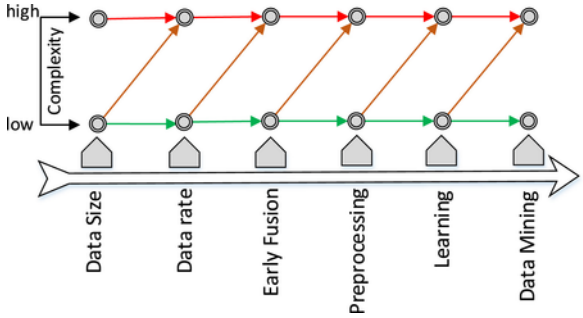

Fig. 10. Factors affecting computational complexity.

Since MDSM applications execute using multiple computing platforms, therefore, application logic is distributed among different devices and systems. For static applications, the data acquisition, and light-weight data stream mining components are used in mobile devices however heavy weight data processing and knowledge aggregation components are installed in remote and resourceful environments. In the case of dynamic application execution, application components are mapped into graph data structures, some optimization operations are performed before the distribution of application logic. However, such techniques lack in MDSM relevant literature. The distribution of application logic in MECC systems is a complex task because edge servers are tightly coupled with infrastructure based clouds and application components reside at all three levels i.e. mobile devices, edge servers, and infrastructure based clouds. These tightly bounded applications are highly dependent over Internet connectivity, therefore, mobility of devices requires continuous virtual machine migrations and tracings of application states. Existing literature still lacks a fully functional MECC based system to facilitate MDSM applications. In addition, alternate solutions are needed to handle the mobility and Internet dependency and tight coupling issues in MECC systems.

Computation offloading strategies help in the partial execution of MDSM applications in distributed computing settings such as F2F, far-edge to the edge, MCC, and MECC communication models. Although there exists numerous computation offloading schemes for general applications, however, existing literature still lacks in data stream mining application specific strategies. Existing literature exhibits the static, dynamic, and adaptive computation offloading strategies which work at method, thread, program, component, and application levels, however, these techniques do not consider the speed and volume of the data stream. In addition, existing computation offloading schemes are either device or cloud-centric, or work in collaboration between mobile and cloud systems however new methods are required to offload data stream mining tasks in MECC systems.

Since MDSM applications need to handle continuous data streams in dynamically changing mobile environments, therefore, data management strategies are needed for raw data and/or partially processed data. Existing MDSM platforms either collect and store raw data streams using separate applications or process raw data streams immediately after collection. Offline data collection strategies increases the latency and online data stream processing enforce the light-weight execution of data mining algorithms. Therefore, new data management strategies are needed which can handle maximum data stream online but use heavy-weight algorithms for data processing. Alternately, strategies are required to manage the partially processed transient data streams.

Finally, existing MDSM systems primarily manage the knowledge patterns in cloud environments. In addition, a few systems exist which manage the knowledge locally using on-board storage and performs on-demand knowledge synchronization between mobile and cloud environments. However, new knowledge management methods are required for MECC based MDSM applications.

8.3. Technical research challenges

In addition with above mentioned challenges, next-generation MDSM applications and platforms need to handle following technical research challenges.

8.3.1. Multi-tier architectures

The computing technologies are growing rapidly and next-generation MDSM platforms needs to use these processing technologies in order to accelerate the application performance. Despite of wide acceptance existing literature still lacks the multi-tier and heterogeneous data processing platforms. Therefore, future MDSM platforms should be designed with scalable topological settings using heterogeneous computing architectures blended with CPUs, GPUs, FPGAs, and large scale data centers. In addition, hierarchical memory architectures based on Caches, RAMs, and internal and external storage should also considered to design next-generation applications and platforms.

8.3.2. Load-balancing

Considering the advancements in computing technologies, future MDSM platforms will span across resource-constrained IoTs, wearable, and mobile devices at one end and resourceful servers, clusters, and multi-cloud infrastructures on the other end. Future MDSM platforms need to integrate efficient load-balancing strategies in order to minimize the latency, efficient energy utilization, reduce bandwidth consumption and in-network data movement across the platforms. The new load-balancing strategies may integrate fuzzy logic and soft set theory based methods for improved efficiency. In addition, deep context models could be used in order to improve the load-balancing strategies across the platforms.

8.3.3. Optimization

The streaming data in mobile environments challenges the capacities of MDSM platforms in terms of energy consumption, storage management, bandwidth utilization, performance gain, privacy preservation, scheduling, and workflow management. Considering the above mentioned challenges, the MDSM applications and platforms need to be optimized for data processing, task scheduling, privacy preservation, and knowledge management. In addition with this the optimization algorithms should ensure seamless application execution across multiple devices and computing systems. The MDSM platforms should enable dynamic and adaptive application execution in MECC systems. TO further the research, the optimization strategies should be devised to achieve the maximum trade-off between data processing efforts and application execution in multiple platforms. Considering the optimization objectives, new algorithms must ensure the reduced and optimal resource consumption both for application execution and the resource required to execute the optimization algorithms itself.

8.3.4. Data stream and knowledge management

MDSM applications need to handle the data streams in multiple formats and need different data management strategies. The MDSM platforms must provide the optimal data management schemes for raw data streams. To this end, existing in-memory data management schemes needs to be improved in order to efficiently handle the streaming data considering its velocity, variety, volume, and variability characteristics. MDSM applications convert raw data streams into different formats at each stage of execution. These formats include raw data converted into event data streams, feature vectors, structured formats such as tables, to name a few. In addition, the intermediate data generated during data processing, when the data populated in data structures (i.e. arrays, trees, and graphs), challenge the computational capacities of resource constrained devices and computing systems which have low amount of available memory. New data management strategies are required to efficiently handle the intermediate data streams. Finally, the MDSM applications produce knowledge patterns which need to be integrated and summarized for a holistic view of incoming data streams. Future MDSM platforms must provide synchronized knowledge management schemes across the MECC systems.

8.3.5. Programming models, design patterns, and development environments

Considering the heterogeneity in next-generation MDSM applications and platforms, new programming models, design patterns, and development environments are needed. Existing simulation tools and programming models support application execution as either mobile-

first or cloud-first approach, however, new programming models should support the application execution across MECC systems. In addition, new design patterns are required which could be reused to each the application development process in MECC systems. Moreover, new integrated development environments (IDEs) are needed to integrate the programming models and design patterns. The IDEs should provide support for drop and drop component based visual workflow management across MECC systems. Further, the IDEs should provide reusable components for rapid application development in MECC systems.

8.4. Future research areas

This section presents some future directions (see Fig. 11) in order to accelerate the research work in MDSM applications and platforms. Due to application and platform level heterogeneity, MDSM applications can help in future and emerging research areas in multiple ways.

8.4.1. Privacy and security

The onboard data sources in far-edge devices produce personal data streams, therefore, MDSM applications need to address the privacy and security concerns of end users (Sokolova and Matwin, 2016). However, to this end, existing literature lacks in scalable end-to-end privacy preservation models for MDSM applications in mobile edge cloud computing systems (Chang et al., 2016). The privacy preservation models are needed to be designed and embedded in existing MDSM applications without loosing the quality of uncovered knowledge patterns. Moreover, the data stream mining applications should enable secure data and knowledge transfer strategies for data movement inside MDSM platforms. To this end, privacy and security challenges need serious attentions in order to prevail this important research area.

8.4.2. Big data reduction

The continuous evolution in mobile data streams eventually results in big data. However, analyzing the massive amount of data and uncovering useful patterns for end users is a challenging task. The deployment of data stream mining applications at user-end can help in reducing big data wherein the users can uncover the knowledge patterns using personal far-edge devices. The resultant knowledge patterns could be shared in order to reduce big data (Rehman and Batool, 2015). Existing literature lacks the pattern based data sharing strategies for big data systems. Future research work should focus on the development and deployment of learning models complying with the needs of big data systems. In addition, pattern sharing, knowledge summarization, and big data aggregation models are needed in order to deal with reduced big data. In essence pattern based big data reduction can benefit to users and big data system providers in many ways including, (a) reduced data communication cost, (b) minimum bandwidth utilization, (c) reduced in-network data movement, (d) fewer efforts in data cleaning and preprocessing for conversion of unstructured big data in to structured datasets, and last but not the least, (e) big data system providers can offer personalized services to end users.

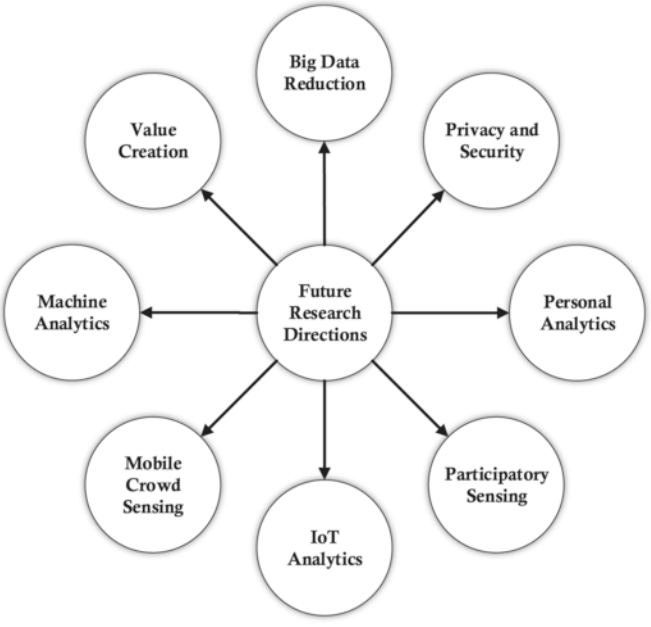

Fig. 11. Future research areas.

8.4.3. Value creation

MDSM applications in MECC systems can help in value creation for customers and enterprises in multiple ways. At one end, the customers can use the personal far-edge devices, edge servers, and cloud computing systems to find the personal knowledge patterns. At the other end, enterprises can acquire the customers' data in order to develop and optimize their business process models and meet their needs (Chang, 2014). MDSM applications can benefit in value creation for a wide spectrum of user-centered business models such as that used for e-commerce, personalized health and insurance, tourism, Telecom, amongst others.

8.4.4. Machine analytics

MDSM applications can benefit in machine analytics in order to uncover the operating and performance behaviors of machines. The embedded data stream mining components in machines can help in onboard and off-board data collection and uncovering machine behaviors in MECC systems. For example, in manufacturing industries, large scale industrial production units can use embedded data stream mining components to uncover knowledge patterns from machine log files and monitor the machine's performance. Similarly, local and collective intelligence in robotics can be embedded using MDSM applications. Few more example applications include smart cars, vehicular ad-hoc networks, machine to machine communication systems, and cyber-physical systems.

8.4.5. Personal analytics

Mobile users generate personal data from a plethora of sensory and non-sensory data sources (Rehman et al., 2015). These data sources collect data streams of mobile users from onboard and off-board sensors as well as the data generated in the result of user interactions with mobile devices, physical activities performed by users, and the behavioral data of users on social networks and World Wide Web. MDSM applications in MECC system can help in uncovering personal knowledge patterns from above mentioned personal data. The knowledge patterns are useful for lifestyle and wellness management applications, behavioral analytic driven systems, mobile health applications, mobile social networks, and mobile commerce, to name a few.

8.4.6. IoT analytics

MDSM applications can be embedded in IoT systems in order to uncover the device-centric and collective knowledge patterns (Satyanarayanan et al., 2015). The applications can be deployed in a single device and multi-device settings. In single device settings, the uncovered knowledge patterns could be used for improving single device usage experiences however in the case of multi-device settings, the patterns could be used for the overall improvement of IoT systems. In addition, the application logic could be distributed across multiple IoT devices in order to find the collective behavior.

8.4.7. Mobile crowd sensing

The MDSM applications in MECC systems can facilitate in mobile crowd sensing systems (Jayaraman et al., 2014a). For example, the data streams collected by smart city management applications for traffic management, commuters facilitation, crowd management in sporting arenas, and facilitating pilgrims and peoples gatherings at holy places. Similarly, MDSM applications can facilitate in management crowds of animals, vehicles, IoTs and many more similar applications.

8.4.8. Participatory sensing

Participatory sensing is another application area for MDSM applications and platform. The knowledge patterns generated by mobile users can help governments, business, enterprises, corporations, and third party public data stream collectors in order to develop user-driven applications and systems. However, participatory sensing systems must ensure user privacy and security of shared data. In addition, new incentive mechanisms are needed in order to lure mobile users for participatory data sharing.

In this section, we discussed a few future research areas for the intervention of MDSM applications and platforms. However, the tremendous growth in IoTs, big data, cloud computing, and mobile edge computing has risen many new application areas and research opportunities for MDSM applications and platforms. Therefore, we perceive that using MDSM in MECC system will quickly prevail in all sectors of the economy and humane lifestyle management.

9. Conclusion

MDSM applications execute in multiple topological settings in multiple phases. Each phase of MDSM applications need to handle heterogeneity which increases the computational complexity. MDSM applications are deployed in different computing devices and systems with different form factors. Therefore MDSM systems need to enable multiple functionalities such as application partitioning, computation offloading, data management, light-weight and heavy-weight data processing, knowledge management, and adaptation strategies to name a few. Existing literature review reveals that MDSM applications need to handle six critical factors to handle complexity namely, (a) size of data stream, (b) speed of data, (c) early data fusion, (d) selection of preprocessing methods, (e) learning model development strategies, and (f) selection of data mining algorithms. Therefore future research work must focus these factors in order to optimize MDSM applications to achieve multiple objectives such as, (a) efficient energy utilization, (b) optimal bandwidth utilization, (c) reduced data movement in MECC systems, (d) achieving memory efficiency, and (e) performance enhancement in terms of latency and CPU usage.

Acknowledgements

The work presented in this article is supported by the Ministry of Education Malaysia (FRGS FP051-2013A and UMRG RP001F-13ICT) and Bright Spark Unit of University of Malaya for providing incentive support. The authors also extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for its funding this prolific research group (PRG-1436-16).

References

Samsung unveils galaxy s5 and new gear range. (2014, February) Online. [Online]. Available: http://www.samsung.com/uk/discover/mobile/samsung-unveils-galaxy-s5-and-new-gear-range/〉

Android (operating system). (2016, 03) [Online]. Available: 〈https://en.wikipedia.org/wiki/Android(operatingsystem)〉

Apple iphone history. (2016, 03) [Online]. Available: 〈http://apple-history.com/iPhone〉

Web of science databases. (2016, 03) online. [Online]. Available: 〈www.webofknowledge.com〉

Abdallah, Z.S., Gaber, M.M., Srinivasan, B., Krishnaswamy, S., 2012. Cbars: Cluster based classification for activity recognition systems. In: Advanced Machine Learning Technologies and Applications. Springer, 2012, pp. 82–91.

Abdallah, Z.S., Gaber, M.M., Srinivasan, B., Krishnaswamy, S., 2015. Adaptive mobile activity recognition system with evolving data streams. Neurocomputing 150, 304–317.

Abolfazli, S., Sanaei, Z., Ahmed, E., Gani, A., Buyya, R., 2014. Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges. IEEE Commun. Surv. Tutor. 16 (1), 337–368.

Agrawal, R., Srikant, R., et al., 1994. Fast algorithms for mining association rules. In: Proceedings 20th Int. Conf. Very Large Data Bases, VLDB, vol. 1215, pp. 487–499.

Ahmad, A., Ahmad, E., 2016. A survey on mobile edge computing. In: Proceedings of the 10th Internatonal Conference on Intelligenct Systems and Control (ISCO), Coimbatore, India doi: 10.1109/ISCO.2016.7727082.

Ahmad, R.W., Gani, A., Hamid, S.H.A., Shiraz, M., Yousafzai, A., Xia, F., 2015. A survey on virtual machine migration and server consolidation frameworks for cloud data centers. J. Netw. Comput. Appl. 52, 11–25.

Altomare, A., Cesario, E., Comito, C., Marozzo, F., Talia, D., 2013. Using clouds for smart city applications. In: 2013 IEEE Proceedings of the 5th International Conference on Cloud Computing Technology and Science (CloudCom), vol. 2. IEEE, pp. 234–237.

Arunkumar, S., Srivatsa, M., Rajarajan, M., 2015. A review paper on preserving privacy in mobile environments. J. Netw. Comput. Appl. 53, 74–90.

Bahl, V., (2015, May) The emergence of micro datacenters (cloudlets) for mobile computing. [Online]. Available: 〈http://research.microsoft.com/apps/video/default.aspx?Id=246447〉

Bonomi, F., Milito, R., Zhu, J., Addepalli, S., 2012. Fog computing and its role in the internet of things. In: Proceedings of the First Edition of the mcc Workshop on Mobile Cloud Computing. ACM, pp. 13–16.

Boukhechba, M., Bouzouane, A., Bouchard, B., Gouin-Vallerand, C., Giroux, S., 2015. Online prediction of peoples next point-of-interest, concept drift support. In: Human Behavior Understanding. Springer, pp. 97–116.

Braojos, R., Beretta, I., Constantin, J., Burg, A., Atienza, D., 2014. A wireless body sensor network for activity monitoring with low transmission overhead. In: 2014 Proceedings of the 12th IEEE International Conference on Embedded and Ubiquitous Computing (EUC). IEEE, pp. 265–272.

Chang, V., 2014. The business intelligence as a service in the cloud. Future Gener. Comput. Syst. 37, 512–534.

Chang, V., Kuo, Y.-H., Ramachandran, M., 2016. Cloud computing adoption framework: a security framework for business clouds. Future Gener. Comput. Syst. 57, 24–41.

Chen, F., Deng, P., Wan, J., Zhang, D., Vasilakos, A.V., Rong, X., 2015. Data mining for the internet of things: literature review and challenges. Int. J. Distrib. Sens. Netw. 12 doi: 10.1155/2015/431047.

Cisco, 2015. Cisco visual networking index: Global mobile data traffic forecast update, 2015–2020 (white paper), Tech. Rep., 2015. [Online]. Available: 〈http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html〉

Cord, M., Cunningham, P., 2008. Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval. Springer Science & Business Media doi: 10.1007/978-3-540-75171-7.

Dogan, N., Tanrikulu, Z., 2013. A comparative analysis of classification algorithms in data mining for accuracy, speed and robustness. Inf. Technol. Manag. 14 (2), 105–124.

Donohoo, B.K., Ohlsen, C., Pasricha, S., Xiang, Y., Anderson, C., 2014. Context-aware energy enhancements for smart mobile devices. IEEE Trans. Mob. Comput. 13 (8), 1720–1732.

Dou, A.J., Kalogeraki, V., Gunopulos, D., Mielikinen, T., Tuulos, V., Foley, S., Yu, C., 2011. Data clustering on a network of mobile smartphones. In: 2011 IEEE/IPSJ Proceedings of the 11th International Symposium on Applications and the Internet (SAINT). IEEE, pp. 118–127.

Drolia, U., Martins, R.P., Tan, J., Chheda, A., Sanghavi, M., Gandhi, R., Narasimhan, P. 2013. The case for mobile edge-clouds. In: 2013 IEEE Proceedings of the 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC) Ubiquitous Intelligence and Computing. IEEE, pp. 209–215.

Eom, H., Figueiredo, R., Cai, H., Zhang, Y., Huang, G., 2015. Malmos: Machine learning-based mobile offloading scheduler with online training. In: 2015 Proceedings of the 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud). IEEE, pp. 51–60.

Fernando, N., Loke, S.W., Rahayu, W., 2013. Mobile cloud computing: a survey. Future Gener. Comput. Syst. 29 (1), 84–106.

Ferreira, H., Duarte, S., Preguiça, N., 2010. 4sensing–decentralized processing for participatory sensing data. In: 2010 IEEE Proceedings of the 16th International Conference on Parallel and Distributed Systems (ICPADS). IEEE, 2010, pp. 306–313.

Framework, A., 2015. Allseen alliance . [Online]. Available: 〈https://allseenalliance.org/framework/documentation/learn〉

Fuqiang, Y., 2011. The research on distributed data stream mining based on mobile agent. Procedia Eng. 23, 103–108.

Gaber, M.M., Gama, J., Krishnaswamy, S., Gomes, J.B., Stahl, F., 2014a. Data stream mining in ubiquitous environments: state-of-the-art and current directions. Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 4 (2), 116–138.

Gaber, M.M., Stahl, F., Gomes, J.B., 2014. Pocket data mining framework. In: Pocket Data Mining. Springer, 2014b, pp. 23–40.

Gaber, M.M., Zaslavsky, A., Krishnaswamy, S., 2005. Mining data streams: a review. Sigmod Rec. ACM 34 (2), 18–26.
Gaber, M.M., Zaslavsky, A., Krishnaswamy, S., 2009. Data stream mining. In: Data Mining and Knowledge Discovery Handbook. Springer, pp. 759–787.
Gama, J., 2013. Data stream mining: the bounded rationality. Informatica 37 (1).
Goel, A.M. Mangla, N., Patel, R., 2010. A survey on distributed mobile database and data mining. In: International Conference on Methods and Models in Science and Technology (ICM2ST-10), vol. 1324, no. 1. AIP Publishing, 2010, pp. 207–210.
Goldberg, A.B., Zhu, X., Singh, A., Xu, Z., Nowak, R., 2009. Multi-manifold semi-supervised learning.
Gomes, J.B., Krishnaswamy, S., Gaber, M.M., Sousa, P.A., Menasalvas, E., 2012b. Mars: a personalised mobile activity recognition system. In: Mobile Data Management (MDM), 2012 IEEE Proceedings of the 13th International Conference on, July 23-26, Bangluru, India. IEEE, 2012, pp. 316–319.
Gomes, J.B., Krishnaswamy, S., Gaber, M.M., Sousa, P.A., Menasalvas, E., 2012. 2012a. Mobile Activity Recognition using Ubiquitous Data Stream Mining. Springer, 2012 doi: 10.1007/978-3-642-32584-7_11.
Gu, T., Wang, L., Wu, Z., Tao, X., Lu, J., 2011. A pattern mining approach to sensor-based human activity recognition. IEEE Trans. Knowl. Data Eng. 23 (9), 1359–1372.
Ha, K., Chen, Z., Hu, W., Richter, W., Pillai, P., Satyanarayanan, M., 2014. Towards wearable cognitive assistance. In: Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services. ACM, pp. 68–81.
Ha, K., Satyanarayanan, M., 2015. Openstack++ for Cloudlet Deployment. School of Computer Science Carnegie Mellon University Pittsburgh.
Haghighi, P.D., Krishnaswamy, S., Zaslavsky, A., Gaber, M.M., Sinha, A., Gillick, B., 2013. Open mobile miner: a toolkit for building situation-aware data mining applications. J. Organ. Comput. Electron. Commer. 23 (3), 224–248.
Hassan, M.A., Wei, Q., Chen, S., 2015. Elicit: Efficiently identify computation-intensive tasks in mobile applications for offloading. In: 2015 IEEE International Conference on Networking, Architecture and Storage (NAS). IEEE, pp. 12–22.
Hromic, H., Le Phuoc, D., Serrano, M., Antonic, A., Zarko, I.P., Hayes, C., Decker, S., 2015. Real time analysis of sensor data for the internet of things by means of clustering and event processing. In: 2015 IEEE International Conference on Communications (ICC). IEEE, pp. 685–691.
Huang, G., Song, S., Gupta, J.N., Wu, C., 2014. Semi-supervised and unsupervised extreme learning machines. IEEE Trans. Cybern. 44 (12), 2405–2417.
Jayaraman, P.P., Gomes, J.B., Nguyen, H.L., Abdallah, Z.S., Krishnaswamy, S., Zaslavsky, A., 2014a. Cardap: a scalable energy-efficient context aware distributed mobile data analytics platform for the fog. In: Advances in Databases and Information Systems. Springer, pp. 192–206.
Jayaraman, P.P., Perera, C., Georgakopoulos, D., Zaslavsky, A., 2014b. Mosden: a scalable mobile collaborative platform for opportunistic sensing applications. arxiv:1405.5867.
Kargupta, H., 2016. Vehicle data mining based on vehicle onboard analysis and cloud-based distributed data stream mining algorithm. February 4, 2016, uS Patent 20,160,035,152.
Kargupta, H., Gilligan, M., Puttagunta, V., Sarkar, K., Klein, M., Lenzi, N., Johnson, D., 2010. Minefleet: the vehicle data stream mining system for ubiquitous environments. In: Ubiquitous Knowledge Discovery. Springer, pp. 235–254.
Khan, A.M., Lee, Y.-K., Lee, S., Kim, T.-S., 2010. Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis, In: 2010 Proceedings of the 5th International Conference on Future Information Technology (FutureTech), IEEE, 2010, pp. 1–6.
Khan, A.M., Siddiqi, M.H., Lee, S.-W., 2013. Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones. Sensors 13 (10), 13099–13122.
Khan, M.A., 2015. A survey of computation offloading strategies for performance improvement of applications running on mobile devices. J. Netw. Comput. Appl. 56, 28–40.
Krishnaswamy, S., Gama, J., Gaber, M.M., 2012. Mobile data stream mining: from algorithms to applications, In: 2012 IEEE Proceedings of the 13th International Conference on Mobile Data Management (MDM). IEEE, 2012, pp. 360–363.
Kumar, K., Liu, J., Lu, Y.-H., Bhargava, B., 2013. A survey of computation offloading for mobile systems. Mob. Netw. Appl. 18 (1), 129–140.
Li, J., Peng, Z., Xiao, B., Hua, Y., 2015. Make smartphones last a day: Pre-processing based computer vision application offloading, In: 2015 Proceedings of the 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), IEEE, pp. 462–470.
Liang, Y., Zhou, X., Yu, Z., Guo, B., 2014. Energy-efficient motion related activity recognition on mobile devices for pervasive healthcare. Mob. Netw. Appl. 19 (3), 303–317.
Lin, C., Choy, K.-l., Pang, G., Ng, M.T., 2013. A data mining and optimization-based real-time mobile intelligent routing system for city logistics, In: 2013 Proceedings of the 8th IEEE International Conference on Industrial and Information Systems (ICIIS), IEEE, pp. 156–161.
Liu, J., Ahmed, E., Mhiraz, M., Gani, A., Buyya, R., Qureshi, A., 2015. Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions. J. Netw. Comput. Appl. 48, 99–117.
Liu, P., Chen, Y., Tang, W., Yue Q., 2012. Mob ile weka as data mining tool on a ndroid. In: Advances in Electrical Engineering and Automation. Springer, pp. 75–80.
Lu, H., Frauendorfer, D., Rabbi, M., Mast, M.S., Chittaranjan, G.T., Campbell, A.T., Gatica-Perez, D., Choudhury, T., Stresssense: Detecting stress in unconstrained acoustic environments using smartphones, In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing. ACM, 2012, pp. 351–360.
Luan, T.H., Gao, L., Li, Z., Xiang, Y., Sun, L., 2015. Fog computing: Focusing on mobile users at the edge, arXiv preprint arXiv:1502.01815.
Martens, J., 2010. Deep learning via hessian-free optimization, In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, Haifa, Israel, pp. 735–742.
Min, J.-K., Cho, S.-B., 2011. Activity recognition based on wearable sensors using selection/fusion hybrid ensemble, In: 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2011, pp. 1319–1324.
Mukherji, A, Srinivasan, V., Welbourne, E., 2014. Adding intelligence to your mobile device via on-device sequential pattern mining, In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication. ACM, pp. 1005–1014.
Murphy, R.R., 2016. Emergency informatics: using computing to improve disaster management. Computer 49 (5), 19–27.
Nguyen, H.-L., Woon, Y.-K., Ng, W.-K., 2015. A survey on data stream clustering and classification. Knowl. Inf. Syst. 45 (3), 535–569.
Oneto, L., Ghio, A., Ridella, S., Anguita, D., 2015. Learning resource-aware classifiers for mobile devices: from regularization to energy efficiency. Neurocomputing 169, 225–235.
Ortiz, J., Huang, C.-C., Chakraborty, S., 2015. Get. more less: Real.-Time Image Cl ust. Mob. phones 2015.(arXiv preprint arXiv:1512.02972)
Oshin, T.O., Poslad, S., Zhang, Z., 2015. Energy-efficient real-time human mobility state classification using smartphones. IEEE Trans. Comput. 64 (6), 1680–1693.
Parthasarathy, S., Ghoting, A., Otey, M.E., 2007. A survey of distributed mining of data streams. Data Streams. Springer. pp. 289–307.
Pasricha, S., Donohoo, B.K., Ohlsen, C., 2015. A middleware framework for application-aware and user-specific energy optimization in smart mobile devices. Pervasive Mob. Comput. 20, 47–63.
Rai, A., Yan, Z., Chakraborty, D., Wijaya, T.K., Aberer, K., 2012. Mining complex activities in the wild via a single smartphone accelerometer, In: Proceedings of the Sixth International Workshop on Knowledge Discovery from Sensor Data. ACM, pp. 43–51.
Rehman, M. H., Batool, A. The Concept of Pattern based Data Sharing in Big Data Environments. *Int. J. Data. Th. App.*, 2015 vol. *8*(4), 11-18.
Rehman, M.H., Chang, V., Batool, A., Teh, Y.W., et al., 36(6, Part A), 917 - 928. doi: 10.1016/j.ijinfomgt.2016.05.013, 2016. Big data reduction framework for value creation in sustainable enterprises. Int. J. Inf. Manag.
Rehman, M.H., Khan, A.R., Batool, A., 2016a, . Big data analytics in mobile and cloud computing environments. Innov. Res. Appl. -Gener. High. Perform. Comput.349–367. (IGI Global) doi: 10.4018/978-1-5225-0287-6.ch014.
Rehman, M.H., Liew, C.S., Iqbal, A., Wah, T.Y., Jayaraman, P.P., 2016c. Opportunistic computation offloading in mobile edge cloud computing environments. In: Proceedings of the 17th IEEE International Conference on Mobile Data Management, Porto, Portugal, (Vol. 1, pp. 208–213). doi: 10.1109/MDM.2016.4013–17
Rehman, M.H., Liew, C.S., Wah, T.Y., 2014. Frequent pattern mining in mobile devices: A feasibility study, In: 2014 International Conference on Information Technology and Multimedia (ICIMU), IEEE, 2014, pp. 351–356.
Rehman, M.H., Liew, C.S., Wah, T.Y., Shuja, J., Daghighi, B., 2015. Mining personal data using smartphones and wearable devices: a survey. Sensors 15 (2), 4430–4469.
Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N., 2009. The case for vm-based cloudlets in mobile computing. IEEE Pervasive Comput. 8 (4), 14–23.
Satyanarayanan, M., Simoens, P., Xiao, Y., Pillai, P., Chen, Z., Ha, K., Hu, W., Amos, B., 2015. Edge analytics in the internet of things, IEEE Pervasive Computing, no. 2, pp. 24–31.
Settles, B., 2012. Active learning. Synth. Lect. Artif. Intell. Mach. Learn. 6 (1), 1–114.
Sharma, S., Chang, V., Tim, U.S., Wong, J., Gadia, S., 2016. Cloud-based emerging services systems. Int. J. Inf. Manag.
Sherchan, W., Jayaraman, P.P., Krishnaswamy, S., Zaslavsky, A., Loke, S., Sinha, A., 2012. Using on-the-move mining for mobile crowdsensing, In: 2012 IEEE Proceedings of the 13th International Conference on Mobile Data Management (MDM). IEEE, pp. 115–124.
Shoaib, M., Bosch, S., Incel, O.D., Scholten, H., Havinga, P.J., 2014. Fusion of smartphone motion sensors for physical activity recognition. Sensors 14 (6), 10146–10176.
Sidek, K.A., Mai, V., Khalil, I., 2014. Data mining in mobile ecg based biometric identification. J. Netw. Comput. Appl. 44, 83–91.
Siirtola, P., Röning, J., 2012. Recognizing human activities user-independently on smartphones based on accelerometer data. Int. J. Interact. Multimed. Artif. Intell. 1 (5).
Siirtola, P., Roning, J., 2013. Ready-to-use activity recognition for smartphones, In: 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM). IEEE, pp. 59–64.
Simoens, P., Xiao, Y., Pillai, P., Chen, Z., Ha, K., Satyanarayanan, M., 2013. Scalable crowd-sourcing of video from mobile devices, In: Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services. ACM, pp. 139–152.
Sokolova, M., Matwin, S., 2016. Personal privacy protection in time of big data. In: Challenges in Computational Statistics and Data Mining. Springer, pp. 365–380.
Srinivasan, V., Moghaddam, S., Mukherji, A., Rachuri, K.K., Xu, C., Tapia, E.M., 2014. Mobileminer: Mining your frequent patterns on your phone, In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, pp. 389–400.
Stahl, F., Gaber, M.M., Aldridge, P., May, D., Liu, H., Bramer, M., Philip, S.Y., 2012. Homogeneous and heterogeneous distributed classification for pocket data mining. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems V. Springer, pp. 183–205.

Suarez-Tangil, G., Tapiador, J.E., Peris-Lopez, P., Pastrana, S., 2015. Power-aware anomaly detection in smartphones: an analysis of on-platform versus externalized operation. Pervasive Mob. Comput. 18, 137–151.
Swan, M., 2012. Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0. J. Sens. Actuator Netw. 1 (3), 217–253.
Talia, D., Trunfio, P., 2010. Mobile data mining on small devices through web services. Mob. Intell. 69, 264.
Tan, G.W.-H., Lee, V.-H., Wong, C.-H., Ooi, K.-B., 2016. Mobile Shopping: the New Retailing Industry in the 21st Century.
Triguero, I., García, S., Herrera, F., 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowl. Inf. Syst. 42 (2), 245–284.
Tsai, C.-W., Lai, C.-F., Chiang, M.-C., Yang, L.T., 2014. Data mining for internet of things: a survey. IEEE Commun. Surv. Tutor. 16 (1), 77–97.
Wang, L., Gu, T., Tao, X., Lu, J., 2012. A hierarchical approach to real-time activity recognition in body sensor networks. Pervasive Mob. Comput. 8 (1), 115–130.
Wang, N., Merrett, G.V., Maunder, R.G., Rogers, A., 2013. Energy and accuracy trade-offs in accelerometry-based activity recognition. In: Proceedings of the 22nd International Conference on Computer Communications and Networks (ICCCN), IEEE, pp. 1–6.
Wickramasinghe, A., Ranasinghe, D.C., 2013. Recognising Activities in Real Time Using Body Worn Passive Sensors with Sparse Data Streams: to Interpolate Or Not to Interpolate?. In *proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services on 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services* (pp. 21-30). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
Wu, P., Zhu, J., Zhang, J.Y., 2013. Mobisens: a versatile mobile sensing platform for real-world applications. Mob. Netw. Appl. 18 (1), 60–80.
Yang, Z., Shangguan, L., Gu, W., Zhou, Z., Wu, C., Liu, Y., 2014. Sherlock: micro-environment sensing for smartphones. IEEE Trans. Parallel Distrib. Syst. 25 (12), 3295–3305.
Ye, F., Ganti, R., Dimaghani, R., Grueneberg, K., Calo, S., 2012. Meca: mobile edge capture and analysis middleware for social sensing applications. In: Proceedings of the 21st International Conference on World Wide Web. ACM, pp. 699–702.
Yoon, J., 2013. Three-tiered data mining for big data patterns of wireless sensor networks in medical and healthcare domains. In: Proceedings of the 8th International Conference on Internet and Web Applications and Services. Rome, Italy, pp. 23–28.
Yuan, B., Herbert, J., 2014. A cloud-based mobile data analytics framework: Case study of activity recognition using smartphone, In: 2014 Proceedings of the 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud). IEEE, pp. 220–227.
Zhang, H., Chen, G., Ooi, B.C., Tan, K.-L., Zhang, M., 2015. In-memory big data management and processing: a survey. IEEE Trans. Knowl. Data Eng. 27 (7), 1920–1948.

Muhammad Habib ur Rehman is a Ph.D. student at FCSIT, UM. He completed his M.Sc. from COMSATS Institute of Information Technology, Pakistan. He is working on data stream mining systems for Internet of Things. His research covers a wide spectrum of application areas including smart cities, mobile social networks, Quantified self, and mHealth. The key research areas of his interest are: mobile computing, edge-cloud computing, Internet of Things, and mobile distributed analytics.(https://sites.google.com/site/drmhr2017/home)

Chee Sun LIEW completed his Masters of Science (Computer Science), in Distributed Computing and Networks from University Sains Malaysia in 2002. He holds a Ph.D. in Informatics from the University of Edinburgh, under Malaysia Ministry of Higher Education scholarship program. His Ph.D. research was related to workflow optimization, under the supervision of Prof Malcolm Atkinson, former UK e-Science envoy, and contributed to the European funded projects on big data and e-Science. He is now working as a senior lecturer in the Faculty Of Computer Science and Information Technology.

Ying-Wah Teh received his B.Sc. and M.Sc. from Oklahoma City University and Ph.D. from University of Malaya. He is currently an Associate Professor at Information Science Department, faculty of Computer Science and Information Technology, University of Malaya. His research interests include data mining and text mining.

Muhammad Khurram Khan is currently working as a Full Professor at the Center of Excellence in Information Assurance (CoEIA), King Saud University, Kingdom of Saudi Arabia. He is one of the founding members of CoEIA and has served as the Manager R&D from March 2009 to March 2012. He developed and successfully managed the research program of CoEIA, which transformed the center as one of the best centers of research excellence in Saudi Arabia as well as in the region.

Prof. Muhammad Khurram Khan is the Editor-in-Chief of a well-esteemed SCI-indexed international journal 'Telecommunication Systems' published by Springer-Verlag since 1993. Furthermore, he is the full-time Editor/Associate Editor of several ISI-indexed international journals/magazines, including IEEE Communications Magazine, Journal of Network & Computer Applications (Elsevier), IEEE Access Journal, Security & Communication Networks (Wiley), IEEE Consumer Electronics Magazine, Journal of Medical Systems (Springer), PLOS ONE (USA), Computers & Electrical Engineering (Elsevier), IET Wireless Sensor Systems, Electronic Commerce Research (Springer), Scientific World Journal (Hindawi), Journal of Computing & Informatics, Journal of Information Hiding and Multimedia Signal Processing (JIHMSP), International Journal of Biometrics (Inderscience), Journal of Physical & Information Sciences, and Journal of Independent Studies and Research-Computing (JISR), etc.

He has also played role of the guest editor of several international ISI-indexed journals of Springer-Verlag and Elsevier Science, etc. Moreover, he is one of the organizing chairs of more than 5 dozen international conferences and member of technical committees of more than 10 dozen international conferences. In addition, he is an active reviewer of many international journals.

Prof. Khurram is an adjunct professor at Fujian University of Technology, China and an honorary Professor at IIIRC, Shenzhen Graduate School, Harbin Institute of Technology, China. He has secured an outstanding leadership award at IEEE international conference on Networks and Systems Security 2009, Australia. He has been included in the Marquis Who's Who in the World 2010 edition. Besides, he has received certificate of appreciation for outstanding contributions in 'Biometrics & Information Security Research' at AIT international Conference, June 2010 at Japan.

He has been awarded a Gold Medal for the 'Best Invention & Innovation Award' at 10th Malaysian Technology Expo 2011, Malaysia. Moreover, his invention recently got a Bronze Medal at '41st International Exhibition of Inventions' at Geneva, Switzerland in April 2013. In addition, he was awarded best paper award from the Journal of Network & Computer Applications (Elsevier) in Dec. 2015.

Prof. Khurram is the recipient of King Saud University Award for Scientific Excellence (Research Productivity) in May 2015. He is also a recipient of King Saud University Award for Scientific Excellence (Inventions, Innovations, and Technology Licensing) in May 2016.

Prof. Khurram has published over 260 research papers in the journals and conferences of international repute. In addition, he is an inventor of 10 US/PCT patents. He has edited 7 books/proceedings published by Springer-Verlag and IEEE. He has secured several national and international research grants in the domain of information security. His research areas of interest are Cybersecurity, digital authentication, biometrics, multimedia security, and technological innovation management. Prof. Khurram has recently played a leading role in developing 'BS Cybersecurity Degree Program' and 'Higher Diploma in Cybersecurity' at King Saud University.

He is a Fellow of the IET (UK), Fellow of the BCS (UK), Fellow of the FTRA (Korea), senior member of the IEEE (USA), a member of the IEEE Technical Committee on Security & Privacy, and a member of the IEEE Cybersecurity community.