Knowl Inf Syst (2019) 58:249–273
https://doi.org/10.1007/s10115-018-1183-0

SURVEY PAPER

Data summarization: a survey

Mohiuddin Ahmed[1]

Received: 4 April 2017 / Revised: 5 January 2018 / Accepted: 14 March 2018 /
Published online: 21 March 2018

Abstract Summarization has been proven to be a useful and effective technique supporting data analysis of large amounts of data. Knowledge discovery from data (KDD) is time consuming, and summarization is an important step to expedite KDD tasks by intelligently reducing the size of processed data. In this paper, different summarization techniques for structured and unstructured data are discussed. The key finding of this survey is that not all summarization techniques create a summary suitable for further analysis. It is highlighted that sampling techniques are a viable way of creating a summary for further knowledge discovery such as anomaly detection from summary. Also different summary evaluation metrics are discussed.

Keywords Summarization · Structured data · Unstructured data · Machine learning · Statistics · Semantics · Natural language processing · Cyber security

1 Introduction

Summarization is a process of creating concise, yet informative, version of the original data. The terms concise and informative are quite generic and depend on application domains. Summarization has been extensively studied in many domains including text analysis, network traffic monitoring, financial domain, health sector and many others. The summary definition or utility is dependent on the purpose of using it; for example, usage of a text summary and network traffic summary are different. Text summary helps a reader to get the gist of a large amount of text e.g., an essay, whereas the network traffic summary is helpful for a network administrator to understand what is happening in the network.

The concept of summarization is distinct from 'compression'. Both summarization and compression create a concise version of the given data, however, there are important dif-

✉ Mohiuddin Ahmed
m.ahmed.au@ieee.org

[1] Department of ICT and Library Studies, Canberra Institute of Technology, Reid, Australia

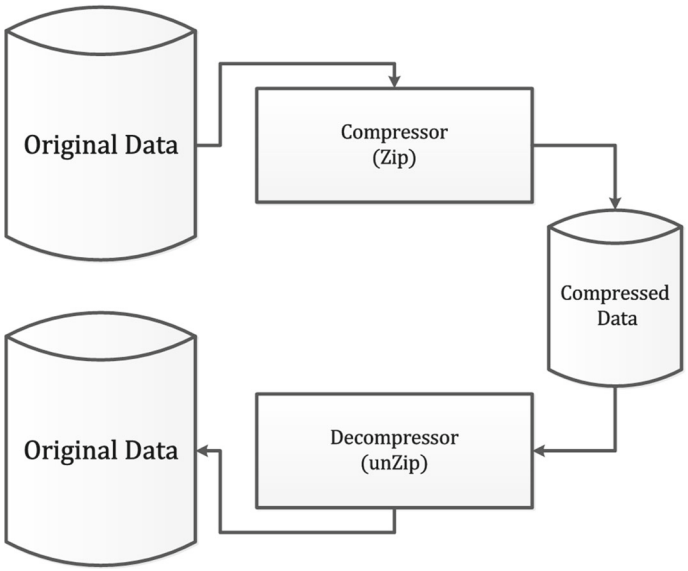

Fig. 1 The conceptual view of data compression

ferences. *Compression* is a syntactic method for reducing the data. Syntactic compression techniques use statistic- or dictionary-based methods and consider the data as large string of bytes [1]. *Compression* makes transporting, emailing, downloading and storing data and software faster and more efficient [2]. A compressed file can be decompressed to recover the original file. Figure 1 shows the conceptual view of data *compression*.

In contrast, summarization uses the semantic content of the data. The concise version of the data via summarization is comprehensive enough without the need for *desummarization* or any inverse operation like unzip to get back to the original data. For example, if there is a large Microsoft excel file, a summarization process creates a concise yet informative version of that file which is another excel file but now only portraying important information from the original file. Therefore, a conclusion can be drawn that, the concept of summarization is different than the compression. In a nutshell, *compression* makes the data non-intelligible and summarization makes the data intelligible for further data analysis and decision making.

Summarization has been proven to be a useful and effective data analysis technique for interpreting large amount of data. Knowledge discovery from data (KDD) is time consuming. Figure 2 (Hoplaros et al. [3]) illustrates how summarization is an important step to expedite KDD tasks by intelligently reducing the size of processed data.

The evolution of computer networks has greatly exacerbated computer security concerns, particularly in today's networking environment and advanced computing facilities. Internet Protocols were not designed with security in mind, consequently there are a large variety of intrusion attempts made by both individuals and large botnets [4] on computer networks. Therefore, the detection of network attacks has become the highest priority today. In 2016, Australia invested $220 Million on cyber security research and development to become a cyber smart nation [5]. Anomaly detection is an important data analysis task which is useful for identifying cyber attacks [6]. Although an anomaly is defined by researchers in various ways, based on the application domain [7–22], one widely accepted definition is that of Hawkins [7]: "An anomaly is an observation which deviates so much from other observations

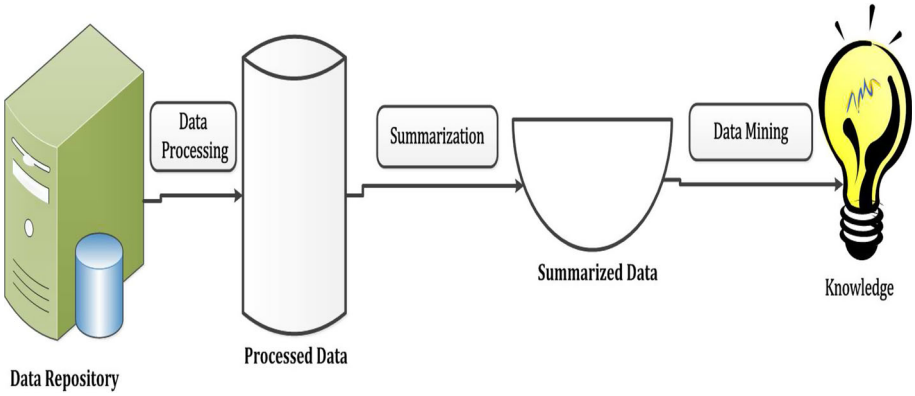

Fig. 2 The process of knowledge discovery in databases (KDD)

as to arouse suspicions that it was generated by a different mechanism". For example, an unusual traffic pattern in a network could mean that a computer has been compromised and data is transmitted to unauthorized destinations. Anomaly detection has been widely applied in countless application domains such as medical and public health, fraud detection, intrusion detection, industrial security, image processing, sensor networks, robots behavior and astronomical data [23–26].

One of the research challenges associated with network anomaly detection is to accurately identify anomalies from huge amounts of data. When data size increases, the anomaly detection techniques perform poorly, due to increasing false alarms and computational cost [23]. However, it is an undeniable fact that, we are awash in a flood of data today. Coffman et al. [27] envisage that Internet traffic doubles each year and Parkinson's law states that, as long as there is storage, data will keep expanding [28]. In today's Internet era, around 40% of the world population are connected and this is one of the reasons for abundance of global IP traffic [29]. From year 2000 to 2016, the number of Internet users have increased more than eight times [29]. Annual global IP traffic will pass the zettabyte[1] threshold by the end of 2016, and is expected to increase to 2 zettabytes by 2019.

Due to the aforementioned statistics, network anomaly detection has been more challenging than it was before. In today's networked environment, the impact of even a limited period of network disruption is high for an organization. Moreover, the network manager needs to extract insights from a large volume of network traffic and take necessary actions. Hence, network managers are only interested in a good summary of network traffic. Consequently, summarization has already been recognized as an important capability in network management [3,30–34].

In order to illustrate the network traffic summarization, an example of a set of network traffic instances in Table 1 is used. The network attributes as shown in Table 1 are the *Source IP address, Destination IP address*, the *protocol* used, and the *source* and *destination ports* used for protocols. The label for each instance shows whether it is a normal or anomalous behavior. In the context of analyzing practical network traffic, such labels are not available. The example in Table 1 is just a very small sample of a much larger set of network traffic instances.

[1] 1 zettabyte is 1000 exabytes and 1 exabyte refers to 1 billion gigabytes.

Table 1 A sample of network traffic

No.	Label	Source IP	Destination IP	Source port	Destination port	Protocol
1	Normal	192.168.5.10	192.168.12.1	20	80	TCP
2	Normal	192.168.5.12	192.168.11.1	21	80	TCP
3	Normal	192.168.12.28	192.168.1.11	22	21	TCP
4	Normal	192.168.5.22	192.168.12.20	23	443	TCP
5	Normal	192.168.12.32	192.168.1.2	25	80	TCP
6	Normal	192.168.5.26	192.168.1.1	53	21	TCP
7	Normal	88.34.224.2	192.168.1.2	110	443	TCP
8	Normal	88.36.226.2	192.168.1.1	119	25	TCP
9	Normal	88.34.226.12	192.168.1.2	143	21	TCP
10	Normal	192.168.5.10	192.168.1.1	443	80	TCP
11	Normal	192.168.5.10	192.168.12.1	20	80	TCP
12	Anomaly	192.168.5.12	192.168.11.1	21	80	TCP
13	Normal	192.168.12.28	192.168.1.11	22	21	TCP
14	Normal	192.168.5.22	192.168.12.20	23	443	TCP
15	Normal	192.168.12.32	192.168.1.2	25	80	TCP
16	Normal	192.168.5.26	192.168.1.1	53	21	UDP
17	Normal	88.34.224.2	192.168.1.2	110	443	ICMP
18	Normal	88.36.226.2	192.168.1.1	119	25	TCP
19	Normal	88.34.226.12	192.168.1.2	143	21	TCP
20	Normal	192.168.5.10	192.168.1.1	443	80	TCP
21	Normal	192.168.5.10	192.168.12.1	20	80	TCP
22	Normal	192.168.5.12	192.168.11.1	21	80	TCP
23	Normal	192.168.12.28	192.168.1.11	22	21	TCP
24	Anomaly	192.168.5.22	192.168.12.20	23	443	ICMP
25	Normal	192.168.12.32	192.168.1.2	25	80	TCP
26	Normal	192.168.5.26	192.168.1.1	53	21	TCP
27	Normal	88.34.224.2	192.168.1.2	110	443	TCP
28	Normal	88.36.226.2	192.168.1.1	119	25	TCP
29	Normal	88.34.226.12	192.168.1.2	143	21	TCP
30	Normal	192.168.5.10	192.168.1.1	443	80	TCP
31	Normal	192.168.5.10	192.168.12.1	20	80	TCP
32	Normal	192.168.5.12	192.168.11.1	21	80	TCP
33	Normal	192.168.12.28	192.168.1.11	22	21	TCP
34	Normal	192.168.5.22	192.168.12.20	23	443	TCP
35	Normal	192.168.12.32	192.168.1.2	25	80	TCP
36	Anomaly	192.168.5.26	192.168.1.1	53	21	TCP
37	Normal	88.34.224.2	192.168.1.2	110	443	TCP
38	Normal	88.36.226.2	192.168.1.1	119	25	TCP
39	Normal	88.34.226.12	192.168.1.2	143	21	TCP
40	Normal	192.168.5.10	192.168.1.1	443	80	TCP

Table 2 Qualitative comparison among recent surveys

Data	Survey						
	1	2	3	4	5	6	7
Structured	×	×	×	×	×	×	✓
Unstructured	✓	✓	✓	✓	✓	✓	✓
Application	**Survey**						
	1	2	3	4	5	6	7
Cyber security	×	×	×	×	×	×	✓
Visualization	×	×	×	×	×	✓	✓
Big data	×	×	×	×	✓	×	✓
Text processing	✓	✓	✓	✓	✓	✓	✓
Evaluation	**Survey**						
	1	2	3	4	5	6	7
Metrics	✓	×	✓	×	×	×	✓

1—Sherif et al. [37], 2—Gambhir et al. [38], 3—Das et al. [39], 4—Nenkova et al. [40], 5—Hesabi et al. [41], 6—Liu et al. [36], 7—[this survey]

Providing meaningful summaries of network traffic is a task which is far from trivial due to the volume and complexity of the data [35]. Given the constantly changing types and mixture of services that are active in the Internet, it is impractical to define the patterns of interest *apriori*. For example, given a set of network traffic, a summary needs to be created that will include the anomalous instances along with all other types of traffic instances.

1.1 Contributions of this survey

In the literature, there is a lack of surveys on data summarization. Very recently, Liu et al. [36] contributed a survey on graph data. Few others focused on text data summarization. The following Table 2 reflects a qualitative comparison among the existing surveys and this survey. It is reflecting that the majority of the existing surveys are suitable for natural language processing and visualization. However, this survey is a comprehensive overview of the data summarization methods and covers a number of critical applications. Additionally, only two of the existing surveys cover the evaluation metrics for summarization. Different summarization techniques and their effectiveness of summarization techniques for knowledge discovery tasks are discussed. Additionally, different summary evaluation metrics are also discussed along with two new metrics named as 'Anomaly Representability' and 'Type Informativity' for evaluating network traffic summary.

1.2 Organization

The rest of the paper is organized as follows. Section 2 describes the role of data summarization along with a taxonomy. Sections 3 and 4 contain discussion on summarization of unstructured and structured data, respectively. Section 5 discusses data stream summarization techniques followed by detailed discussion on the summary evaluation metrics in Sect. 6. The paper concludes with open research problems in Sect. 7.

Table 3 Typical rates at which network traffic instances grow for different network sizes, assuming a packet size of 100 bytes, adapted from [35]

Network size	Link speed	Packet rate	Data hour	Window size (4 TB) (h)
Small	100M ethernet	125 kp/s	25 GB/h	160
Medium	Gigabit ethernet	1.25 Mp/s	300 GB/h	13
Large	OC-192	12 Mp/s	2.5 TB/h	1.5

2 The role of data summarization

Intelligent analysis of data is a challenging task in many domain. In reality, the volume of the datasets is quite high and the time required to perform data analysis, such as anomaly detection, increases with data size. It has been shown that, a summary of the large data is easier and faster to analyze [3,31–35,42]. Summarization has been widely explored in many domains, including transactional databases, network data streams, intrusion detection systems (IDS), point of sales data (POS) and natural text [3]. For example, in computer networks, a network administrator needs to monitor the activity of the network. Even for a small company network, the amount of data generated from different network applications (e.g., email, http and p2p applications) is huge and cannot be analyzed easily [3]. Table 3 shows the typical rate at which network traffic accumulates for various link speeds. It is evident that, even for a small network the traffic data accumulated over one hour is too much to be analyzed manually. Consequently, a summary of the network traffic is very helpful for network managers to quickly assess what is happening in the network (Table 3).

Consider the following scenario. A network manager wants to know about the different types of anomalous events that occurred in the network in the past month. The actual size of the data containing anomalous traffic of month's data is intractable for human analysis. If anomaly detection is applied on the whole month's data, computation will take a long time. Summarizing the data makes it smaller while retaining the key characteristics of the data; this allows easier human analysis as well as less computational time for anomaly detection techniques.

Knowledge discovery from data is a process that can sometimes be very time consuming. In the case of anomaly detection as a knowledge discovery task, research [32–35,42] has shown that it is possible to detect anomalies from the summary. Consequently, summarization can be a preprocessing step before performing anomaly detection (as shown in Fig. 3) on the large data.

2.1 Taxonomy of summarization

The problem of summarization arises in the context of a variety of data analysis tasks and application domains. For example, market basket data analysis requires a summary to identify the shopping pattern of customers. TV news readers are given a summary to deliver the news, which requires document summarization. In order to support these tasks, a variety of summarization techniques have been developed. Figure 4 shows a taxonomy of the summarization approaches where the techniques are divided into two major categories as for structured and unstructured data. In the following sections, an overview of different approaches of summarization techniques are provided.

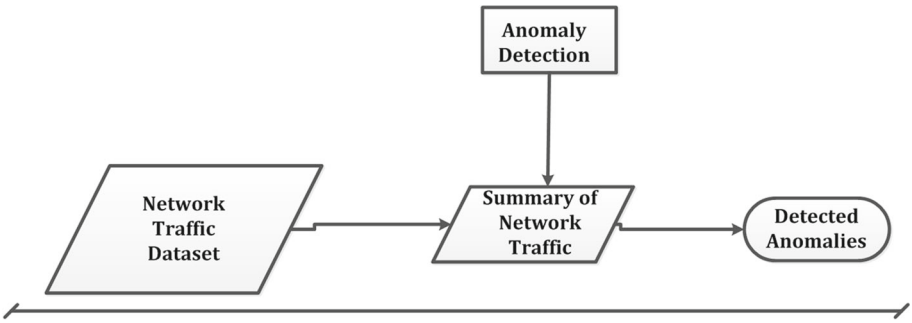

Fig. 3 The role of data summarization for network traffic analysis

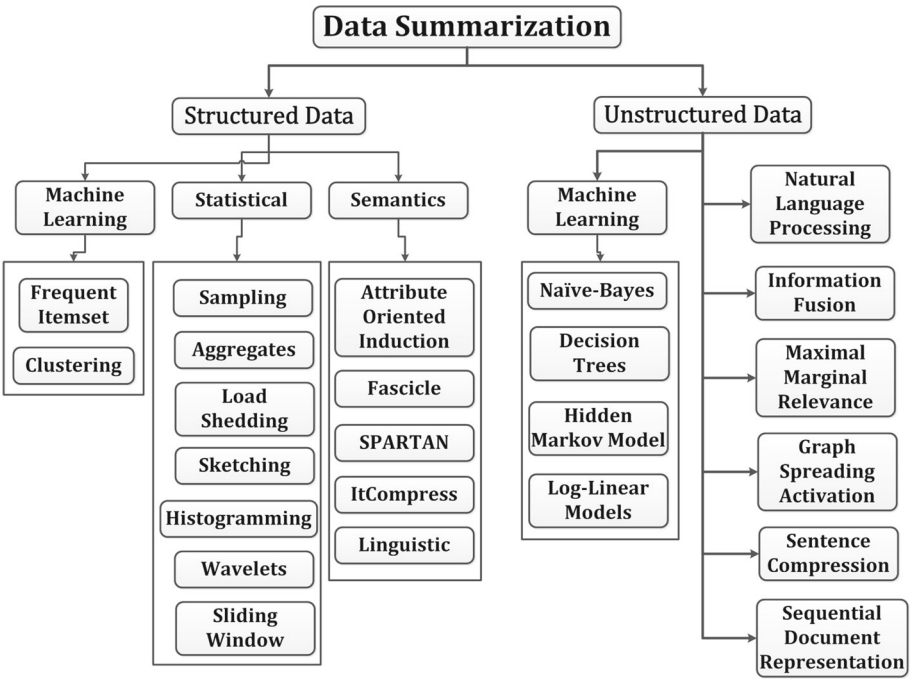

Fig. 4 Taxonomy of data summarization

3 Summarization of unstructured data

Unstructured data refers to information that either does not have a pre-defined data model or is not organized in a pre-defined manner. Unstructured data is typically text-heavy, but may also contain dates and numbers. In the context of summarization, unstructured data is considered as text, therefore, summarization of unstructured data is essentially text summarization. Automated text summarization has been an established research domain for almost the last half century. Dipanjan et al. [39] provided a survey on text summarization covering majority of the research efforts made in the past. Radev et al. [43] defined text summarization as "A text that is produced from one or more texts, that conveys important information in the

original text(s), and that is no longer than half of the original text(s) and usually significantly less than that". Text summarization is the combination of the following processes [43]:

- **Extraction**: Finds the key phrases or sentences and produces a summary.
- **Abstraction**: Produces the key information in a new way.
- **Fusion**: Extracts important parts from the text and combines them coherently.
- **Compression**: Discards irrelevant or unimportant text.

Different summarization techniques have been proposed by researchers for single and multiple text documents. An early work, Luhn et al. [44] proposed a method based on the frequency of any particular word, which is an useful measure to identify its significance. Baxendale et al. [45] emphasized using the position in the document to find the important sentences in the text. Machine Learning (ML) approaches for text summarization started in the 1990s. Some of the major ML techniques are discussed as follows:

- **Naive–Bayes classifier:** Kupiec et al. [46] derived a classification function that was able to learn from the text and could categorize each sentence as significant or not. Edmundson et al. [47] improved it by using criteria like presence of uppercase words and sentence length in a Naive–Bayes classifier. Aone et al. [48] also developed an automatic text summarization system called *DimSum* using a Naive–Bayes classifier, but used richer properties such as inverse document frequency and term frequency, etc.
- **Decision tree:** Text documents usually follow an anticipated discourse structure, and important sentences tend to exist in certain specifiable positions such as the abstract, the title, etc. This idea was first studied by Lin et al. [49] by giving weights to sentences according to their positions which was termed by them as *positioning method*. In a later work [50] by the same authors, they avoided the independence assumption of features and modeled the sentence extraction problem as decision tree instead of Naive–Bayes classifier.
- **Hidden Markov model (HMM):** Conroy et al. [51] used Hidden Markov Model (HMM) to solve the sentence extraction problem. In contrast to Naive–Bayes methods discussed above, HMM considers the local dependencies between sentences. However, these features used for HMM may not always create meaningful summary.
- **Artificial neural network (ANN):** A popular machine learning technique called artificial neural network (ANN) [52] had been also used in automatic text summarization by Svore et al. [53], Lin et al. [54] and many others.
- **Natural language processing (NLP):** Deep natural language analysis refers to the techniques where important words, phrases, and concepts are identified in a sentence or/and in a single or multiple documents by exploring purely linguistic semantics. Barzilay et al. [55] proposed a text summarization method which used linguistic analysis. The concept of lexical chain (i.e., a sequence of related words in a document that may have short span or long span) is introduced for summarization [55]. The process has three steps:

  1. segmentation of the text,
  2. identification of lexical chains,
  3. using strong lexical chains to identify important sentences for extraction.

- **Similarity measure-based:** A wide range of similarity measures between pairs of sentences had been used for summarization. Some researchers used clustering to identify common themes and represented each cluster by a single sentence [56]. Others represented a cluster by generating a composite sentence [57]. Other approaches used maximal marginal relevance to detect the novelty of current passage with respect to previous

included passages and incorporated only novel passages into the summary [58]. Some recent work has considered the approach for a multilingual environment [59].
– **Topic modeling:** The automatic process of summarizing documents is a major rule in many text applications. Automatic text summarization tries to retain the essential information without affecting the document quality. In [60], a multi-document summarization method that combines topic model and fuzzy logic is proposed. This method extracts relevant topic words from source documents and then extracted words are used as elements of fuzzy sets.

4 Summarization of structured data

Structured data refers to any data that resides in a fixed field (rows and columns) within a matrix or file. This includes data contained in relational databases and spreadsheets. Network traffic data is an example of structured data as it contains data in a number of rows and columns, where each row corresponds to a data instance (although raw network traffic data contains both structured headers and unstructured payload, the structured portion of the data is considered in this paper). In this section, the summarization techniques covering statistical, linguistic and machine learning methods are discussed.

4.1 Statistical techniques

4.1.1 Aggregates

Using aggregates it is possible to estimate the statistical distribution of data that could be utilized to approximate the pattern in the set of data. For example, clustering very large databases is resource intensive specifically in terms of memory and computation time. To handle this type of scenario, a variety of techniques are proposed that employ certain data compression methods. These techniques generally use some statistical functions aggregates, mean, L_1 and L_2 norms to represent the data into a suitable compressed representation. Next, it applies clustering algorithms to the compressed representation to produce final clustering. The statistical functions are able to estimate the characteristics of the data from which they are computed, therefore, they can be used to produce clustering over the whole dataset.

For example, BIRCH [61] or Balanced Iterative Reducing and Clustering is a statistic-based clustering algorithm which builds a dynamic hierarchical tree structure to maintain summary information about candidate clusters. This tree is called Clustering Feature tree (CF-tree) that hierarchically organizes the clusters residing at the leaf nodes. A clustering algorithm is applied to the nodes of CF-tree for the resultant clusters. BIRCH suffers from the fact that they can only use partitioning algorithms such as *k-means* [62] in their subsequent phases when producing final clustering results. Hierarchical algorithms cannot be used because they are based on distances between data objects which are not well represented by distances between compressed objects (CF-tree nodes), specifically when compression is very high. This problem is solved by [63–66] suggesting specialized compressed representative objects called *data bubbles* which are more suitable for hierarchical clustering.
Clearly, aggregates are only defined for numerical values so above mentioned summarization approaches are feasible only for numerical datasets. The limitation of these approaches is that they cannot handle dataset containing categorical values or other types of attributes.

4.1.2 Sampling

A sample is a subset of the dataset. Sampling is a powerful tool used to handle large databases for KDD. When it is infeasible to apply any KDD process to a dataset because of its size, a sample is drawn from the dataset, and the KDD process is applied to that sample and the result is generalized to the whole dataset. Sampling is a popular choice for reduction of input data in data mining and machine learning techniques. The principal advantages of sampling over complete enumeration are the reduced cost and greater speed. There are different kinds of sampling in practice. Here, the major categories [67] of sampling are briefly discussed as follows.

- **Simple random sampling:** Given the sample size, simple random sampling chooses sample at random where no data instance is included more than once.
- **Stratified random sampling:** The dataset is divided into non-overlapping subsets. These subsets are called *strata*. The sampling scheme selects a random element from each *strata* and produces a stratified sample. Basically, a simple random sampling is applied on each *strata* to have a stratified random sample.
- **Systematic sampling:** In systematic sampling, a data instance is sampled from the dataset, beginning from a specified starting point to the end, at equal intervals. For example, if the first random instance's location is 2 (starting point) and the interval value is 5, then for a sample of size 3, the sample instances are from the 2nd, 7th and 12th locations, respectively. The interval is calculated as rounded up $\left[\frac{Size\ of\ data}{Size\ of\ sample}\right]$.
- **Cluster random sampling:** The whole dataset is organized into groups (clusters); groups are randomly selected according to sampling rate, and all members of the selected groups are selected. For example, divide a school into a number of classes, randomly select few classes, and consider all students in the selected rooms as the sample.
- **Multi-stage random sampling:** The dataset is organized into groups; randomly select groups, and then randomly select members in these groups (an equal number selected per group). This is just an extension of cluster random sampling. For example, repeat the steps for cluster random sampling, finally, randomly select students in each selected class.

4.2 Semantic-based summarization

4.2.1 Linguistic summary

Pouzols [68] proposed an approach to solve the problems of the tools that generate summaries of network traffic flow records. To enhance the human understanding of the network traffic summaries, linguistic summaries based on fuzzy logic are proposed. Linguistic summaries are natural language expressions that describe some important facts about the given data. According to Yager et al. [69], a basic linguistic summary is comprised of three basic components as follows:

- A summary, which can be described using a linguistic expression, semantically represented by fuzzy logic.
- A quantifier to explain the summary.
- A quality measure of the summary.

Although linguistic summaries make it easier to understand the characteristics of the network traffic, these cannot be used for further data analysis task such as anomaly detection.

4.2.2 Attribute oriented induction

Attribute Oriented Induction (AOI) is a concept description approach from descriptive data mining domain, first proposed in 1989 [70]. It is further studied and commercially adopted; first in *DBLearn* [71], and then in *DBMiner* [72]; which is an extension of *DBLearn*. AOI is a generalization process which abstracts a large dataset from low conceptual level to relatively a higher conceptual level. The aim of the AOI process is to describe data in a concise manner and present important and interesting general properties of the dataset. However, AOI suffers from over generalization of attributes. This shortcoming had been studied and AOI approach is extended in [73,74] to alleviate this problem.

4.2.3 Fascicle

Jagadish et al. [75] introduced the notion of fascicles to reduce the size of relational databases for storage minimization. Also, the authors propose a method for extracting patterns from the compressed databases produced. Fascicles rely on an extended form of association rules—more precisely, frequent itemset mining—to achieve data summarization. The authors highlight the fact that, often, many rows in a dataset share similar values for several attributes.

Fascicles are designed to exploit the similarities of rows and group them together to represent them in a concise and compact form as a summary. It performs lossy semantic compression by approximating representative values in fascicles where the degree of approximation is specified by user parameters that guarantees an upper bound on error in approximation.

The ideas developed for fascicles are mostly interesting for handling numerical values since defining an aggregate function for numerical values is intuitive. However, less attention is given in the case of categorical attributes.

4.2.4 SPARTAN

SPARTAN is another semantic-based summarization technique proposed in [76] by extending or generalizing the fascicles approach. Fascicles do not consider the correlation among attributes, but that is emphasized specifically in SPARTAN. The SPARTAN is based on the idea of exploiting predictive data correlations and predetermined error tolerances for individual attributes to construct a concise and accurate Classification and Regression Tree (CaRT) model. The attributes that can be predicted by a Bayesian network [77] using other attributes are then omitted and CaRT model is generated to predict those attributes. Clearly CaRT is a compact model that is used instead of the attributes (predicted attributes) for which it is built as well as predicting attributes.

The approach was designed to maximize the reduction of data tables size for storage purpose only. Even though CaRT gives a high level understanding of attribute dependencies, applications such as mining cannot use the data tables compressed with SPARTAN without a form of decompression beforehand [78].

4.2.5 ItCompress

Iterative Compression or ItCompress was proposed by Jagadish et al. [79] for relational databases. The approach tries to compress a relation **R** by reducing the number of rows by grouping similar rows and representing them by a Representative Row (RR). The idea is,

given a tolerance parameter for each attribute, the algorithm searches for set of RRs which offers the best match for the rows in original relation **D** within the range of tolerance given to it.

The RRs that are found by algorithm are given a unique ID (RRid) and are stored in a separate relation called RR relation. The original relation **D** is then transformed to a compressed relation denoted as D_c which contains three attributes: (1) RR_{id}, (2) Bitmap, and (3) Outlying List. The RR_{id} in D_c is a pointer that matches rows in RR relation. Bitmap is a sequence of binary digits equal to number of attributes in original relation **D**, where a bit is set to 1 if there is a match between the attribute value in the original relation and the representative relation, and 0 otherwise. Similarly, the third attribute Outlying List stores attribute values in **D** that are different than the ones represented by the RRid and the bitmap [80].

4.3 Machine learning

In Machine Learning, supervised and unsupervised learning techniques are two widely used knowledge discovery techniques. Supervised learning is the machine learning task of inferring a function from labeled training data [25,81]. The training data consist of a set of training examples. In supervised learning, each training example consists of an input object and a desired output value. A supervised learning algorithm learns from the training data and creates a knowledge base which can be used for mapping new and unseen data. For example, Support Vector Machine [82] refers to a supervised learning algorithm, where pre-labeled data is required. Labeled data are rare and difficult to find. However, even when pre-labeled data is available, events that are not present in the labeled data, such as zero day attacks in intrusion detection domain, are not handled well [6].

Unsupervised learning tries to find hidden structure in unlabeled data, which distinguishes unsupervised learning from supervised learning [25,81]. For example, Clustering refers to unsupervised learning algorithms, where pre-labeled data is not required to extract rules for grouping similar data instances [25,83].

4.3.1 Frequent itemsets

Intuitively, a set of items that appears more frequently than rest of the items are called "frequent itemsets". To be formal, it is assumed that, there is a support threshold, which is a number to reflect the number of times an item appears in a basket. If I is a set of items in a basket, I can be frequent if its support is equal or more than the threshold. Frequent itemset mining has been used for network traffic summarization by Chandola et al. [31] and Hoplaros et al. [3]. Chandola et al. [31] viewed summarization as the process of compacting a given number of data instances to a smaller set of summary elements, where each summary element represents a subset of the input data in a way that each data instance is represented in the summary. Their Bottom-Up Summarization (BUS) algorithm performs frequent itemset mining on the input dataset, then greedily searches the itemsets, identifying the ones with maximum *conciseness* and minimum *information loss* (the terms conciseness, information loss are defined in Summary Evaluation, Sect. 6 in this paper), and substituting data instances in the itemset by a representative element. This is iterated until the desired conciseness is achieved.

Hoplaros et al. [3] modified the BUS algorithm to use summarization metrics as an objective function. The characteristics of BUS makes its objective function easy to modify and limiting the chances of errors, and allowing it to focus on the applicability of the metrics.

Hoplaros et al. modified the second step of BUS, instead of searching for only the maximum conciseness and minimum information loss, they also used interestingness and intelligibility metrics to create the summary.

4.3.2 Clustering

Clustering is an important unsupervised Machine Learning technique, whose aim is to find natural or intrinsic groups of unlabeled objects so that objects within groups are highly similar and objects across the groups are highly dissimilar [25]. Clustering can also be used for data summarization. Using the concept of clustering, two different summary definitions have been widely used. These are *centroid-based* and *feature-wise intersection-based* summarization. Among a large set of available clustering algorithms [83], the *k-means* algorithm [62] is widely used for summarization due to its simplicity which has the ability to handle high-dimensional data.

In a *Centroid*-based summarization, the dataset is clustered and the cluster centroids are used to form the summary [84]. The *k-means* algorithm has been widely used for this type of summarization [85,86]. For example, Ha-Thuc et al. [84] proposed a quality-threshold data summarization method modifying the *k-means* algorithm, where the summary is the set cluster centroids. The number of clusters is determined using the characteristics of the dataset and a *threshold*. The algorithm partitions a dataset until the distortion or sum of squared error (1) is less than a given *threshold*. It starts by finding the cluster centroids as *k-means* but the next steps are executed only if the distortion is above the given threshold and the existing cluster is split. A new centroid is introduced which is closer to the larger cluster centroid. This process is repeated until every cluster's distortion is below the given threshold. The method to choose the threshold and how the characteristics of datasets are analyzed, have not been addressed [84–86].

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} d(c_i, x)^2; \tag{1}$$

In a *feature-wise intersection*-based summarization [31], once the dataset is clustered, then the summary is created from each cluster using the feature-wise intersection of the data instances. Consequently, summaries from all the clusters are combined together to produce the final summary. This approach is suitable when data instances with a cluster have identical attribute values, for example, if the *protocol* attribute is considered in a sample cluster of network traffic data instances which has similar protocols such as TCP. Chandola [31] stated that the clustering-based approach can produce a good summary because it can capture the frequent modes of behavior but performs poorly when the dataset contains outlying data instances.

5 Data stream summarization

A data stream is an ordered sequence of instances which are generated rapidly. Examples of data streams include computer network traffic, web searches, sensor data and many more. Data streams are also referred to as 'dynamic datasets'. In a streaming paradigm it is not possible to store all the data because of resource constraints. Therefore, stream summarization requires algorithms with memory requirements that are independent of the data size. Most of the

techniques discussed so far do not have this property. In this section, existing data stream summarization techniques are discussed.

5.1 Sampling

Sampling is a simple but effective method for summary construction from data streams. It is also relatively straight forward to use the summary in a wide variety of application since the sampling-based summary representation uses the same multi-dimensional form as the original data. According to Charu [87,88], reservoir-based sampling methods [89] are useful for data stream analysis. Advantages of sampling-based methods for summarization are the following [90]:

- Easy, efficient and usually provides an unbiased estimate of the underlying data with provable error guarantees.
- Since the sampling methods use the original representation of the data, they are easy to use with any data mining applications. In most cases, the error guarantees of sampling methods generalize to the mining behavior of the underlying applications.

Now, the drawbacks of sampling for summarization are as follows:

- It is desirable to pick out a sample of pre-decided size from the stream. The key issue in case of data stream sampling is the unknown size of the data stream. The challenge is how to adopt sampling rate dynamically as data changes over time.
- In order to maintain an unbiased representation of the underlying data, the probability of including a point in random sample should not be fixed in advance, but should change with progression of the data streams

Load shedding is another stream summarization technique similar to sampling. It has been proposed for data streams [91] that aims to drop a portion of the data in a randomized fashion [92]. Load shedding had shown success in approximate query answering, but it cannot be used for all KDD tasks since it could drop a group of sequences that represent interesting patterns.

5.2 Histogram

Another key method for data summarization is using histograms [42,93,94]. In this method, the data are divided along any attribute into a set of ranges or buckets and the count for each bucket is maintained. Thus, the space requirement is defined by the number of buckets in the histogram. A naïve histogram-based summarization technique would discretize the data into partitions of equal length ("equi-width partitioning") and store the frequencies of these buckets. It is relatively easy to use the histogram for answering different kinds of queries such as range queries [95]. A number of strategies are devised for improved query resolution from the histogram [42,93,94].

However, these methods are considered as one-dimensional summarization techniques. There are some multi-dimensional histograms, and a few of them keep a one-dimensional histograms for each dimension based on the attribute value independence assumption (AVI) [96].

5.3 Bloom filter

Bloom filters are designed for set-membership queries of the discrete instances [97]. For example, the query can be, given a particular instance, has it ever occurred in the data

stream?. Bloom filters provide an efficient technique to maintain a summary of the stream, so that this query can be resolved with a probabilistic bound on the accuracy [87]. In a Bloom filter, false positives are possible but not false negatives. In other words, if the Bloom filter reports that an instance does not belong to the steam, then this will always be true.

Bloom filters are able to answer the set-membership query and it is not possible to use the summaries produced for further processing such as anomaly detection. To use Bloom filter the user have to be familiar with data stream before and only then a query can be placed.

5.4 Sliding window

Another technique proposed for data stream summarization is sliding window [98–100]. To produce an approximate answer to a query over data stream, it is very convenient to compute the query over a recent data rather than over past complete history of the data streams [101,102]. For instance, only last week's data could be considered for answering the query and data older than one week is discarded. When the data stream progresses, the data at the end of the window is discarded and new data at the front is added.

5.5 Wavelets

Wavelets are well-known techniques often used in databases for hierarchical data decomposition and summarization [103,104]. The basic idea behind wavelets is to create a decomposition of the data attributes into a set of wavelet functions and basis functions. The property of the wavelet method is that the higher order coefficients of the decomposition illustrate the broad trends in the data, whereas the more localized trends are captured by the lower order coefficients.

In the context of summarization, a wavelet transformation is mostly used to transform the data and then make it possible to construct a compact representation of the data in a transformed domain. A wavelet transformation can summarize the wavelet transformed data by only saving the strongest wavelet coefficient and setting the rest of coefficients to zero. The wavelet-based summarization are mostly used in application domains such as decision support systems, image and signal processing [104].

5.6 Sketch

Sketch allows basic queries in data stream summarization such as point, range, and inner product queries to be approximately answered very quickly. In addition, it can be used to solve several important problems in data streams such as finding quantiles and frequent items [105–107]. The idea of sketches is essentially an extension of the random projection technique to the time series domain [87]. The idea of using this technique for determining representative trends in the time series domain was first observed in [108].

The main disadvantage of sketch is its inability to handle multi-dimensional data streams [87]. Most sketch methods are based on analysis along a single dimensional stream of data instances. Some recent work in [105] provides a few limited methods for multi-dimensional streams, these are not easily extensible for more general problems. This problem is not however, unique to sketch-based methods. Many other summarization methods such as wavelets or histograms can be extended in a limited way to the multi-dimensional case, and do not work well beyond dimensionality of 4 or 5 [90].

5.7 Stream clustering

The problem of clustering is especially significant in the data stream scenario because of its ability to provide a compact summary of the data stream. Clustering of the data stream is often used as a precursor to other applications such as stream classification. Here, few popular stream clustering algorithms are discussed [109–111].

- StreamKM++ [112]: It computes a small weighted sample of the data stream and uses the *k-means*++ algorithm [113] as a randomized seeding technique to choose the first values for the clusters.
- CluStream [114]: It maintains statistical information about the data using micro-clusters. These micro-clusters are temporal extensions of cluster feature vectors. The micro-clusters are stored at snapshots in time following a pyramidal pattern. This pattern allows to recall summary statistics from different time horizons. The pyramidal time frame is used to store micro-clusters that are captured at specific instant time in order to answer the queries of user over different time horizon.
- ClusTree [115]: It is a non-parametric algorithm that can automatically adapt to the speed of the stream. It uses a compact and self-adaptive index structure for maintaining stream summaries. The underlying cluster structure is not suitable for creating summary with original instances as it uses a compact representation of data like micro-cluster [114].
- Den-Stream [116]: It uses dense micro-clusters (called "core-micro-cluster") to summarize clusters with arbitrary shapes. To maintain and distinguish the potential clusters and outliers, this method presents core-micro-cluster and outlier micro-cluster structures. The micro-clusters maintain statistical information about the data locality [110].
- D-Stream [117]: This method maps each input data record into a grid for summarization and computes the grid density. The grids are clustered based on the density. This algorithm uses a density decaying technique to capture the dynamic changes of a data stream.
- CobWeb [118]: This was one of the first proposed incremental methods for clustering data. It uses a classification tree. Each node in a classification tree represents a class (concept) and is labeled by a probabilistic concept that summarizes the attribute-value distributions of objects classified under the node.

The aforementioned clustering algorithms create summaries that do not contain original data instances or similar structures as they use a compact representation of the data instances. Therefore, the summary produced by these algorithms cannot be directly used for further knowledge discovery tasks such as anomaly detection. This also poses difficulties for human analysts that rely on important data characteristics such as their known attribute values.

6 Evaluation metrics

Evaluating a summary is a difficult task because there does not exist an ideal summary for a given document or set of documents. Evaluation by a human varies to a great extent and evaluation is more difficult than producing the summary in many cases. On the other hand, if the summary evaluator is a machine, then a set of standard metric is sufficient to evaluate the summaries. In the summarization research domain, there are some commonly used metrics which are discussed below.

6.1 Evaluation of summaries of unstructured data

– **Human and automatic evaluation:** Lin et al. [119] described and compared various human and automatic metrics to evaluate summaries. They focus on the evaluation procedure used in the Document Understanding Conference 2001 (DUC-2001), where the Summary Evaluation Environment (SEE) interface was used to support the human evaluation part. NIST assessors in DUC-2001 compared manually written ideal summaries with summaries generated automatically by summarization systems and baseline summaries. Each text was decomposed into a list of units (sentences) and displayed in separate windows in SEE. To measure the summaries, assessors stepped through each model unit (MU) from the ideal summaries and marked all system units (SU) sharing content with the current model unit, rating them with scores in the range 1–4 to specify that the marked system units express all (4), most (3), some (2) or hardly any (1) of the content of the current model unit. Grammaticality, cohesion, and coherence were also rated similarly by the assessors. The weighted recall at threshold t (where t ranges from 1 to 4) was defined as

$$Recall_t = \frac{Number\ of\ MUs\ marked\ at\ or\ above\ t}{Number\ of\ MUs\ in\ the\ model\ summary} \tag{2}$$

– **ROUGE:** Lin [54] introduced a set of metrics called Recall-Oriented Understudy for Gisting Evaluation (ROUGE) that have become standards of automatic evaluation of summaries. Let R = $\{r_1,\ldots,r_m\}$ be a set of reference summaries, and s be a summary generated automatically by some system. Let $\Phi_x(\mathrm{X})$ be a binary vector representing the x-grams[2] contained in a document X; the i-th component $\Phi_x^i(\mathrm{X})$ is 1 if the i-th x-gram is contained in X and 0 otherwise. The metric ROUGE-N is a x-gram recall-based statistic that can be computed as follows where $\langle .,. \rangle$ denotes inner product of vectors.

$$ROUGE - N(s) = \frac{\sum_{r \in R} \langle \Phi_x(r), \Phi_x(s) \rangle}{\sum_{r \in R} \langle \Phi_x(r), \Phi_x(r_s) \rangle} \tag{3}$$

– **Information-theoretic evaluation:** Lin et al. [119] proposed an information-theoretic method for automatic evaluation of text summaries. The central idea is to use a divergence measure between a pair of probability distributions, in this case the Jensen–Shannon divergence, where the first distribution is derived from an automatic summary and the second from a set of reference summaries. This approach has the advantage of working both in the single-document and the multi-document summarization scenarios.

However, all of these metrics are useful for text summarization only.

6.2 Evaluation of summaries of structured data

Like unstructured data summary, structured data summary evaluation is also dependent on the application domain and its usage. Here, the state-of-the-art metrics for network traffic summary evaluation are discussed. To simplify the understanding of a good network traffic summary, here in this section, the existing summary evaluation metrics [3] are explained. The existing metrics are the following.

– **Conciseness:** Conciseness expresses how compact a summary (S) is with respect to the original dataset (D). It is the ratio of input dataset size and the summarized dataset size. Then, conciseness is represented in Eq. (4).

[2] In the field of computational linguistics, an x-gram is a contiguous sequence of x items from a given sequence of text or speech.

$$Conciseness = \frac{|S|}{|D|} \tag{4}$$

– **Information loss:** A general metric used to describe the amount of information lost from the original dataset as a result of the summarization. Loss is defined as the number of distinct values not present in the summary [3]. Equation (5) states the information loss of a summary (S), where T is the number of distinct values present in the original data (D) and L defines the difference between number of distinct values present in summary and original data.

$$Information\ Loss = \frac{L}{T} \tag{5}$$

– **Interestingness:** This metric is proposed in [3] which focuses on the objective measures of interestingness with applicability to summarization, emphasizing diversity. Equation (6) defines the interestingness, where n states how many of the data instances in the original dataset are covered by the summary. For example, if the summary of a dataset is a centroid or mean of the data instances, then it indicates that all the data instances are covered by the centroid-based summary and therefore the interestingness becomes one.

$$Interestingness = \frac{n \times (n-1)}{|D| \times (|D|-1)} \tag{6}$$

– **Intelligibility:** This metric is used to measure how meaningful a summary is, based on the attributes present in the summary [3]. Intelligibility is defined and displayed in Eq. (7), a is the number of attributes present in the original dataset and q is the number of attributes present in the summary. For example, if there are 10 attributes ($a = 10$) in the original dataset and a summary only contains 5 of them ($q = 5$) then, *Intelligibility* is 0.5.

$$Intelligibility = \frac{q}{a} \tag{7}$$

6.3 New summarization metrics

In this section, two new summarization metrics to properly evaluate the summarization algorithms are discussed. Among the existing metrics, conciseness reflects how terse the summary is with respect to the original data. The information loss metric provides insight about the presence/absence of the data instances in the summary and has inversely proportional relationship with conciseness. The other two metrics interestingness and intelligibility do not provide much insight about the summary from the anomaly detection point of view. Therefore, a new metric is required; anomaly representability, which is based on the presence of anomaly in the summary. To provide more understanding of the network in the summary, type informativity metric evaluates a summary based on the types of traffic present in the summary and original data. For example, if the summary contains all the types of network traffic such as *www, p2p, ftp, mail, database, multimedia,* etc., which are present in the original data, the *type informativity* score is one. These two metrics are discussed in detail next.

6.3.1 Anomaly representability

When the aim is to create summaries that can be useful for anomaly detection and such summary may contain two kinds of data instances, one belonging to normal behavior and the other belonging to anomalies. The existing summarization metrics discussed earlier are not focussed on identifying anomalies, therefore, cannot evaluate a summary based on the presence of anomalies. However, it is an important aspect of summarization that the summary

contains both normal and anomalous instances which can be used for anomaly detection. The newly devised metric, anomaly representability, that reflects the amount of normal and anomalous data instances present in the summary. Anomaly representability is defined in Eq. (8), where a_S is the number of anomalous data instances in summary S and a_D is the number of anomalous data in the original dataset D. Consequently, higher value of anomaly representability metric refers to a summary's suitability for existing anomaly detection techniques. For example, if the dataset D has 100 anomalous data instances and in the summary S has 30 anomalous data instances. The anomaly representability of the summary S is $\frac{30}{100} = 0.3$. This metric is scaled between 0 and 1.

$$Anomaly\ Representability = \frac{a_S}{a_D} \tag{8}$$

6.3.2 Type informativity

This metric concerns the extent to which the contents of a summary represent all the different types of traffic present in original data, such as *www, p2p, ftp, mail, database, multimedia* etc. A summary should ideally include the different types of behavior, either normal or anomalous. Hence, type informativity evaluates a summary based on the types of traffic present in the summary that are also present in the original data. Type informativity is formally defined in Eq. (9), where $type_S$ is the number of traffic types present in summary, S and $type_D$ is the number of traffic types present in the original dataset D. For example, if the original dataset contains traffic activities: *www, mail, p2p* and the summary contains only *mail*, the type informativity of the summary will be $\frac{1}{3} = 0.33$. Type informativity is scaled between 0 and 1, where 0 means an empty summary and 1 when all the types of traffic are present in the summary.

$$Type\ Informativity = \frac{type_S}{type_D} \tag{9}$$

7 Conclusion

Summarization is very much application specific. Vast majority of the existing surveys on summarization are focussed on text and graph [36–41]. According to [120], 'summary is a text that is produced from one or more text, that conveys important information from the original text, and that is no longer than half of the original text and usually significantly less than that'. Liu et al. [36] stated that, graph summary is a sparsified graph, that reveals patterns from the original data and preserves specific structural properties. However, in this survey, the focus of data summarization is given on cyber security applications (i.e., interesting pattern identification as well as anomaly detection [6]) and therefore covered a wide range of summarization techniques. The key finding is that the summary requirements for cyber security applications are different than the text- and graph-based applications.

Summarization is considered as an important intermediate step for various knowledge discovery tasks. Interesting pattern detection from summary instead of analyzing huge amount of original data is a computationally efficient process. In this paper, the critical analysis of the literature shows that for anomaly detection, sampling-based summarization techniques are suitable than the other techniques. Based on the sampling techniques, new methods are required to create ideal summary for anomaly detection. Additionally, two new summary evaluation metrics for structured data are also devised.

7.1 Open research problems

Based on the characteristics of different summarization techniques discussed in Sects. 3, 4 and 5, a conclusion can be drawn that, only sampling-based techniques are able to create summary for further knowledge discovery tasks such as anomaly detection, since the other techniques do not follow the conditions for the summary to contain parts of original data which may be used. However, only sampling may not contain the rare anomalies in the summary; consequently, if the summary does not contain anomalies then it is useless to apply anomaly detection on that summary. Therefore, it is needed to devise new summarization techniques which can answer the following research questions:

- *How effectively can summarization include interesting patterns in the summary?* When summarization is used as intermediate step for rare anomaly detection, it is important that the summary includes such anomalies. Summarization methods are not all equally effective for this purpose.
- *Can the summarization process automatically determine the appropriate size of the summary?* For interesting pattern identification, an ideal summary should contain anomalies from the original data but, at the same time, the summary needs to be concise. Therefore, identifying the appropriate summary size is an important aspect of the summarization process. Simultaneously, the summary size has impact on the performance of anomaly detection methods in terms of computation time and accuracy.
- *Can summarization reduce the total time of anomaly detection?* Performing anomaly detection on the original data is computationally slow. On the other hand, a much smaller summary can be processed to detect anomalies much faster. However, the time taken to create the summary must be taken into account. It is not clear that the total time for summarization and anomaly detection can improve on the time for anomaly detection on the original data while achieving comparable performance in detecting anomalies.
- *How can summarization work in a streaming scenario?* Most Internet-based applications are streaming in nature. Therefore, data stream processing, where data cannot be stored or revisited [121] is an emerging domain of applications. Certainly heavy network traffic can be considered a stream. Addressing the challenge of summarization in such environment requires different techniques than in a non-streaming environment.

References

1. Salomon D (2006) Data compression: the complete reference. Springer, New York
2. WinZip (2016) Accessed on 07 March 2016
3. Hoplaros D, Tari Z, Khalil I (2014) Data summarization for network traffic monitoring. J Netw Comput Appl 37:194–205
4. Papalexakis EE, Beutel A, Steenkiste P (2012) Network anomaly detection using co-clustering. In: Proceedings of the 2012 international conference on advances in social networks analysis and mining (ASONAM 2012), ASONAM'12, Washington, DC, USA. IEEE Computer Society, pp 403–410
5. The Australian Cyber Security Centre (2016) Accessed on 24 May 2016
6. Ahmed M, Mahmood A, Jiankun H (2015) A survey of network anomaly detection techniques. J Netw Comput Appl 60:19–31
7. Hawkins D (1980) Identification of outliers (monographs on statistics and applied probability), 1st edn. Springer, Berlin
8. Barnett V, Lewis T (1978) Outliers in statistical data, 2nd edn. Wiley, New York
9. Rousseeuw PJ, Leroy AM (1987) Robust regression and outlier detection. Wiley, New York
10. Laurikkala J, Juhola M, Kentala E (2000) Informal identification of outliers in medical data. In: The fifth international workshop on intelligent data analysis in medicine and pharmacology

11. Dantong Y, Sheikholeslami G, Zhang A (2002) Findout: finding outliers in very large datasets. Knowl Inf Syst 4(4):387–412
12. Knorr EM, Ng RT (1998) Algorithms for mining distance-based outliers in large datasets. In: Proceedings of the 24th international conference on very large data bases, VLDB'98, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., pp 392–403
13. Ramaswamy S, Rastogi R, Shim K (2000) Efficient algorithms for mining outliers from large data sets. SIGMOD Rec 29(2):427–438
14. Ghoting A, Parthasarathy S, Otey ME (2008) Fast mining of distance-based outliers in high-dimensional datasets. Data Min Knowl Disc 16(3):349–364
15. Breunig MM, Kriegel H-P, Ng RT, Sander J (2000) Lof: Identifying density-based local outliers. SIGMOD Rec 29(2):93–104
16. Hu T, Sung SY (2003) Detecting pattern-based outliers. Pattern Recogn Lett 24(16):3059–3068
17. Hawkins S, He H, Williams G, Baxter R (2002) Outlier detection using replicator neural networks. In: Kambayashi Y, Winiwarter W, Arikawa M (eds) Data warehousing and knowledge discovery, lecture notes in computer science, vol 2454. Springer, Berlin, pp 170–180
18. Schölkopf B, Platt JC, Shawe-Taylor JC, Smola AJ, Williamson RC (2001) Estimating the support of a high-dimensional distribution. Neural Comput 13(7):1443–1471
19. Aggarwal C, Yu S (2005) An effective and efficient algorithm for high-dimensional outlier detection. VLDB J 14(2):211–221
20. Jagadish HV, Koudas Nick, Muthukrishnan S (1999) Mining deviants in a time series database. In: Proceedings of the 25th international conference on very large data bases, VLDB'99, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., pp 102–113
21. Shekhar S, Chang-Tien L, Zhang P (2003) A unified approach to detecting spatial outliers. GeoInformatica 7(2):139–166
22. Cheng T, Li Z (2006) A multiscale approach for spatio-temporal outlier detection. Trans GIS 10(2):253–263
23. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):15:1–15:58
24. Ahmed M, Mahmood AN, Hu J (2014) Outlier detection, chapter 1. In: Pathan ASK (ed) The state of the art in intrusion prevention and detection. CRC Press, New York, pp 3–21
25. Ahmed M, Mahmood AN, Rafiqul Islam M (2016) A survey of anomaly detection techniques in financial domain. Future Gener Comput Syst 55:278–288
26. Ahmed M, Anwar A, Mahmood AN, Shah Z, Maher MJ (2015) An investigation of performance analysis of anomaly detection techniques for big data in scada systems. EAI Endorsed Trans Ind Netw Intell Syst 15(3):1–16
27. Coffman KG, Odlyzko AM (2002) Internet growth: is there a "Moore's law" for data traffic? In: Abello J, Pardalos PM, Resende MG (eds) Handbook of massive data sets. Kluwer Academic Publishers, Norwell, pp 47–93
28. Kamma D, Geetha G, Neela JP (2013) Countering Parkinson's law for improving productivity. In: Proceedings of the 6th India software engineering conference, ISEC'13, New York, NY, USA. ACM, pp 91–96
29. The Zettabyte Era-Trends and Analysis. Accessed 02 April 2016
30. Ahmed M, Mahmood AN, Maher MJ (2015) An efficient approach for complex data summarization using multiview clustering. In: Jung JJ, Badica C, Kiss A (eds) Scalable information systems. Springer, Cham, pp 38–47
31. Chandola V, Kumar V (2007) Summarization—compressing data into an informative representation. Knowl Inf Syst 12(3):355–378
32. Ahmed M, Mahmood AN, Maher MJ (2015) A novel approach for network traffic summarization. In: Jung JJ, Badica C, Kiss A (eds) Scalable information systems. Springer, Cham, pp 51–60
33. Ahmed M, Mahmood AN, Maher MJ (2015) An efficient technique for network traffic summarization using multiview clustering and statistical sampling. EAI Endorsed Trans Scalable Inf Syst 15(5):1–9
34. Ahmed M, Mahmood AN (2014) Clustering based semantic data summarization technique: a new approach. In: IEEE 9th conference on industrial electronics and applications (ICIEA), 2014, pp 1780–1785
35. Mahmood AN (2008) Hierarchical clustering and summarization of network traffic data. Ph.D. theses, University of Melbourne
36. Liu Y, Dighe A, Safavi T, Koutra D (2016) A graph summarization: a survey. CoRR. arXiv:1612.04883
37. Elfayoumy S, Thoppil J (2014) A survey of unstructured text summarization techniques. Int J Adv Comput Sci Appl 5(7):149–154

38. Gambhir M, Gupta V (2017) Recent automatic text summarization techniques: a survey. Artif Intell Rev 47(1):1–66
39. Das D, Martins AFT (2007) A survey on automatic text summarization. Technical report, literature survey for the language and statistics II course at Carnegie Mellon University
40. Nenkova A, McKeown K (2012) A survey of text summarization techniques. Springer, Boston, pp 43–76
41. Hesabi ZR, Tari Z, Goscinski A, Fahad A, Khalil I, Queiroz C (2015) Data summarization techniques for big data—a survey. Springer, New York, pp 1109–1152
42. Hesabi ZR, Tari Z, Goscinski A, Fahad A, Khalil I, Queiroz C (2015) Data summarization techniques for big data—a survey. In: Khan SU, Zomaya AY (eds) Handbook on data centers. Springer, New York, pp 1109–1152
43. Radev DR, Hovy E, McKeown K (2002) Introduction to the special issue on summarization. Comput Linguist 28(4):399–408
44. Luhn (1958) The automatic creation of literature abstracts. IBM J Res Dev 2(2):159–165
45. Baxendale PB (1958) Machine-made index for technical literature: an experiment. IBM J Res Dev 2(4):354–361
46. Kupiec J, Pedersen J, Chen F (1995) A trainable document summarizer. In: Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR'95, New York, NY, USA. ACM, pp 68–73
47. Edmundson HP (1969) New methods in automatic extracting. J ACM 16(2):264–285
48. Aone C, Okurowski ME, Gorlinsky J, Larsen B (1999) A trainable summarizer with knowledge acquired from robust nlp techniques. In: Mani I, Maybury MT (eds) Advances in automatic text summarization. MIT Press, Cambridge, pp 71–80
49. Lin C-Y, Hovy E (1997) Identifying topics by position. In: Proceedings of the fifth conference on applied natural language processing, ANLC'97, Stroudsburg, PA, USA. Association for Computational Linguistics, pp 283–290
50. Lin C-Y (1999) Training a selection function for extraction. In: Proceedings of the eighth international conference on information and knowledge management, CIKM'99, New York, NY, USA. ACM, pp 55–62
51. Conroy JM, O'leary DP (2001) Text summarization via hidden Markov models. In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, SIGIR'01, New York, NY, USA. ACM, pp 406–407
52. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5(4):115–133
53. Svore K, Vanderwende L, Burges C (2007) Enhancing single-document summarization by combining RankNet and third-party sources. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), Prague, Czech Republic. Association for Computational Linguistics, pp 448–457
54. Lin C-Y (2004) Rouge: a package for automatic evaluation of summaries. In: Moens M-F, Szpakowicz S (eds) Text summarization branches out: proceedings of the ACL-04 workshop, Barcelona, Spain. Association for Computational Linguistics, pp 74–81
55. Barzilay R, Elhadad M (1997) Using lexical chains for text summarization. In: Proceedings of the ACL workshop on intelligent scalable text summarization, pp 10–17
56. Radev DR, Jing H, Budzikowska M (2000) Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In: Proceedings of the 2000 NAACL-ANLP workshop on automatic summarization, NAACL-ANLP-AutoSum'00, Stroudsburg, PA, USA, vol 4. Association for Computational Linguistics, pp 21–30
57. Barzilay R, McKeown KR, Elhadad M (1999) Information fusion in the context of multi-document summarization. In: Proceedings of the 37th annual meeting of the association for computational linguistics on computational linguistics, ACL'99, Stroudsburg, PA, USA. Association for Computational Linguistics, pp 550–557
58. Carbonell J, Goldstein J (1998) The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval, SIGIR'98, Melbourne, Australia. ACM, pp 335–336
59. Evans DK, Mckeown K, Klavans JL (2005) Similarity-based multilingual multi-document summarization. IEEE Trans Inf Theory 49:1–8
60. Lee S, Belkasim S, Zhang Y (2013) Multi-document text summarization using topic model and fuzzy logic. Springer, Berlin, pp 159–168
61. Zhang T, Ramakrishnan R, Livny M (1996) Birch: an efficient data clustering method for very large databases. In: Proceedings of the 1996 ACM SIGMOD international conference on management of data, SIGMOD'96, New York, NY, USA. ACM, pp 103–114

62. MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. In: Cam LML, Neyman J (eds) Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol 1. University of California Press, pp 281–297
63. Breunig MM, Kriegel H-P, Sander J (2000) Fast Hierarchical Clustering Based on Compressed Data and OPTICS. In: Proceedings of 4th European conference on principles of data mining and knowledge discovery, PKDD 2000 Lyon, France, 13–16 Sept 2000. Springer, Berlin, pp 232–242
64. Breunig MM, Kriegel H-P, Krger P, Sander J (2001) Data bubbles: quality preserving performance boosting for hierarchical clustering. In: ACM SIGMOD conference, pp 79–90
65. Zhou J, Sander J (2003) Data bubbles for non-vector data: speeding-up hierarchical clustering in arbitrary metric spaces. In: Proceedings of the 29th international conference on very large data bases, VLDB '03, vol 29. VLDB Endowment, pp 452–463
66. Patra BK, Nandi S (2011) Tolerance rough set theory based data summarization for clustering large datasets. In: Peters JF, Skowron A, Sakai H, Chakraborty MK, Slezak D, Hassanien AE, Zhu W (eds) Transactions on rough sets XIV. Springer, Berlin, Heidelberg, pp 139–158
67. Cochran WG (1977) Sampling techniques, 3rd edn. Wiley, New York
68. Pouzols FM, Lopez DR, Barros AB (2011) Summarization and analysis of network traffic flow records. In: Mining and control of network traffic by computational intelligence, vol 342 of studies in computational intelligence. Springer, Berlin, Heidelberg, pp 147–189
69. Yager RR (1982) A new approach to the summarization of data. Inf Sci 28(1):69–86
70. Cai Y, Cercone N, Han J (1991) Attribute-oriented induction in relational databases. In: Knowledge discovery in databases. AAAI/MIT Press, pp 213–228
71. Han J, Yongjian F, Huang Y, Cai Y, Cercone N (1994) DBLearn: a system prototype for knowledge discovery in relational databases. SIGMOD Rec (ACM Special Interest Group on Management of Data) 23(2):516
72. Han J, Fu Y, Wang W, Chiang J, Gong W, Koperski K, Li D, Lu Y, Rajan A, Stefanovic N, Xia B, Zaiane OR (1996) Dbminer: a system for mining knowledge in large relational databases. In: Proceedings of 1996 international conference on data mining and knowledge discovery, KDD'96. AAAI Press, pp 250–255
73. Han J, Cai Y, Cercone N (1992) Knowledge discovery in databases: an attribute oriented approach. In: Proceedings of the 18th international conference on very large data bases (VLDB'92). Morgan Kaufmann, pp 547–559
74. Han J, Fu Y (1996) Exploration of the power of attribute-oriented induction. In: Advances in knowledge discovery and data mining. AAAI/MIT Press, pp 399–421
75. Jagadish HV, Madar J, Ng RT (1999) Semantic compression and pattern extraction with fascicles. In: Proceedings of the 25th international conference on very large data bases, VLDB'99, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., pp 186–198
76. Shivnath B, Garofalakis M, Rastogi R (2001) Spartan: a model-based semantic compression system for massive data tables. In: International conference on management of data (SIGMOD 2001)
77. Judea P (2000) Causality: models, reasoning, and inference. Cambridge University Press, New York
78. Pham Q-K, Raschia G, Mouaddib N, Saint-Paul R, Benatallah B (2009) Time sequence summarization to scale up chronology-dependent applications. In: Proceedings of the 18th ACM conference on information and knowledge management, CIKM'09, New York, NY, USA. ACM, pp 1137–1146
79. Jagadish HV, Ng RT, Ooi BC, Tung A (2004) Itcompress: an iterative semantic compression algorithm. In: Proceedings of 20th international conference on Data engineering, 2004, pp 646–657
80. Quang-Khai P (2010) Time sequence summarization: theory and applications. Theses, Université de Nantes
81. Mohri M, Rostamizadeh A, Talwalkar A (2012) Foundations of machine learning. MIT Press, Cambridge
82. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2(2):121–167
83. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323
84. Ha-Thuc V, Nguyen D-C, Srinivasan P (2008) A quality-threshold data summarization algorithm. In: Proceedings of IEEE international conference on research, innovation and vision for the future (RIVF), pp 240–246
85. Wendel P, Ghanem M, Guo Y (2005) Scalable clustering on the data grid. In: Proceedings of the 5th IEEE international symposium cluster computing and the grid (CCGrid)
86. More P, Hall LO (2004) Scalable clustering: a distributed approach. Proc IEEE Int Conf Fuzzy Syst 1:143–148
87. Aggarwal C (ed) (2007) Data streams—models and algorithms. Springer, Berlin
88. Aggarwal CC (2006) On biased reservoir sampling in the presence of stream evolution. In: Proceedings of the 32nd international conference on very large data bases, VLDB'06. VLDB Endowment, pp 607–618

89. Vitter JS (1985) Random sampling with a reservoir. ACM Trans Math Softw 11(1):37–57
90. Aggarwal CC, Yu PS (2007) A survey of synopsis construction in data streams. In: CharuC A (ed) Data streams, advances in database systems, vol 31. Springer, Berlin, pp 169–207
91. Tatbul N, Çetintemel U, Zdonik S, Cherniack M, Stonebraker M (2003) Load shedding in a data stream manager. In: Proceedings of the 29th international conference on very large data bases, VLDB '03, vol 29. VLDB Endowment, pp 309–320
92. Tatbul EN (2007) Load shedding techniques for data stream management systems. Ph.D. thesis, Providence, RI, USA. AAI3272068
93. Poosala V, Ganti V, Ioannidis YE (1999) Approximate query answering using histograms. IEEE Data Eng Bull 22:5–14
94. Poosala V, Haas PJ, Ioannidis YE, Shekita EJ (1996) Improved histograms for selectivity estimation of range predicates. In: Proceedings of the 1996 ACM SIGMOD international conference on management of data, SIGMOD'96, New York, NY, USA. ACM, pp 294–305
95. Kooi RP (1980) The optimization of queries in relational databases. Ph.D. thesis, Cleveland, OH, USA. AAI8109596
96. Poosala V, Ioannidis YE (1997) Selectivity estimation without the attribute value independence assumption. In: Proceedings of the 23rd international conference on very large data bases, VLDB'97, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., pp 486–495
97. Broder A, Mitzenmacher M (2004) Network applications of bloom filters: a survey. Internet Math 1(4):485–509
98. Rivetti N, Busnel Y, Mostefaoui A (2015) Efficiently summarizing data streams over sliding windows. In: IEEE 14th international symposium on network computing and applications (NCA), 2015, pp 151–158
99. Babcock B, Datar M, Motwani R, O'Callaghan L (2002) Sliding window computations over data streams. Technical report 2002-25, Stanford InfoLab
100. Babcock B, Datar M, Motwani R, O'Callaghan L (2003) Maintaining variance and k-medians over data stream windows. In: Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems, PODS '03, New York, NY, USA. ACM, pp 234–243
101. Babcock B, Babu S, Datar M, Motwani R, Widom J (2002) Models and issues in data stream systems. In: Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems, PODS '02, New York, NY, USA. ACM, pp 1–16
102. Muthukrishnan S (2005) Data streams: algorithms and applications. Found Trends Theor Comput Sci 1(2):117–236
103. Keim D, Heczko M, Are W (2001) Wavelets and their applications in databases. In: Tutorial notes of ICDE 2001
104. Stollnitz Eric J, Derose Tony D, Salesin David H (1996) Wavelets for computer graphics: theory and applications. Morgan Kaufmann Publishers Inc., San Francisco
105. Cormode G, Muthukrishnan S (2005) An improved data stream summary: the count-min sketch and its applications. J Algorithms 55(1):58–75
106. Alon N, Matias Y, Szegedy M (1996) The space complexity of approximating the frequency moments. In: Proceedings of the 28th annual ACM symposium on theory of computing, STOC'96, New York, NY, USA. ACM, pp 20–29
107. Charikar M, Chen K, Farach-Colton M (2002) Finding frequent items in data streams. In: Proceedings of the 29th international colloquium on automata, languages and programming, ICALP'02, London, UK. Springer, pp 693–703
108. Indyk P, Koudas N, Muthukrishnan S (2000) Identifying representative trends in massive time series data sets using sketches. In: Proceedings of the 26th international conference on very large data bases, VLDB'00, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., pp 363–372
109. Bifet A, Holmes G, Kirkby R, Pfahringer B (2010) Moa: massive online analysis. J Mach Learn Res 11:1601–1604
110. Silva JA, Faria ER, Barros RC, Hruschka ER, de Carvalho ACPLF, Gama J (2013) Data stream clustering: a survey. ACM Comput Surv 46(1):13:1–13:31
111. Alex N, Hasenfuss A, Hammer B (2009) Patch clustering for massive data sets. Neurocomputing 72(7–9):1455–1469
112. Ackermann MR, Märtens M, Raupach C, Swierkot K, Lammersen C, Sohler C (2012) Streamkm++: a clustering algorithm for data streams. J Exp Algorithmics 17:2.4:2.1–2.4:2.30
113. Arthur D, Vassilvitskii S (2007) K-means++: the advantages of careful seeding. In: Proceedings of the 18th annual ACM-SIAM symposium on discrete algorithms, SODA'07, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics, pp 1027–1035

114. Aggarwal CC, Han J, Wang J, Yu PS (2003) A framework for clustering evolving data streams. In: Proceedings of the 29th international conference on very large data bases, VLDB '03, vol 29. VLDB Endowment, pp 81–92
115. Kranen P, Assent I, Baldauf C, Seidl T (2009) Self-adaptive anytime stream clustering. In: 9th IEEE international conference on data mining, 2009, ICDM '09, pp 249–258
116. Cao F, Ester M, Qian W, Zhou A (2006) Density-based clustering over an evolving data stream with noise. In: 2006 SIAM conference on data mining, pp 328–339
117. Li T, Chen Y (2009) Stream data clustering based on grid density and attraction. ACM Trans Knowl Discov Data 3(3):12:1–12:27
118. Fisher DH (1987) Knowledge acquisition via incremental conceptual clustering. Mach Learn 2(2):139–172
119. Lin C-Y, Cao G, Gao J, Nie J-Y (2006) An information-theoretic approach to automatic evaluation of summaries. In: Proceedings of the main conference on human language technology conference of the North American chapter of the association of computational linguistics, HLT-NAACL '06, Stroudsburg, PA, USA. Association for Computational Linguistics, pp 463–470
120. Radev DR, Hovy E, McKeown K (2002) Introduction to the special issue on summarization. Comput Linguist 28(4):399–408
121. Shah Z, Mahmood AN, Barlow M (2016) Computing hierarchical summary of the data streams. In: Bailey J, Khan L, Washio T, Dobbie G, Huang JZ, Wang R (eds) Advances in knowledge discovery and data mining. Springer, Cham, pp 168–179

Mohiuddin Ahmed has more than 5 years of data science and cyber security experience from both academia and industry. Currently, he is working as a lecturer at Canberra Institute of Technology (CIT), Australia. and also involved with CIT Data Strategy Working Group. Mohiuddin is exploring effectiveness of deep learning to solve Big Data problems and actively engaged in Healthcare Cyber Bio-Security projects. Previously, he worked in the area of text mining and predictive analytics in the AI division at MIMOS, Malaysia. In PhD from UNSW, Mohiuddin has made practical and theoretical contributions in Big Data Analytics (Summarization). His research has high impact on financial fraud detection, critical infrastructure protection (IoT, Smart Grid), information security against DoS attacks, FDIA, etc., and health analytics (heart disease diagnosis). He achieved data science certifications from IBM. Specially he is interested in Industry 4.0 and Blockchain. He is also serving as an associate editor of International Journal of Computers and Applications, Taylor & Francis.