

Received March 6, 2021, accepted March 28, 2021, date of publication April 5, 2021, date of current version April 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3071009

# Enabling Efficient and Scalable Service Search in IoT With Topic Modeling: An Evaluation

MOHAMMAD ABDUR RAZZAQUE<sup>1</sup>, (Senior Member, IEEE)

School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough TS1 3BX, U.K.

e-mail: m.razzaque@tees.ac.uk

**ABSTRACT** Service search in IoT's large-scale, heterogeneous and multi-domain services space is a challenging task. It can take the time that may not be acceptable for many IoT applications and requires resources that may not be available in many IoT devices. A categorisation of these services into their application domains can reduce the search space and offer an efficient and scalable service search. Recently, in many fields, such as short text messages and categorisation of IoT service specifications, generative probabilistic models, like Topic modelling, are being used. Generally, IoT service descriptions are short and sparse. Existing work on IoT services categorisation is based on Latent Dirichlet Allocation (LDA), but it does not perform well in short and sparse texts. Also, IoT services categorisation has few specific issues, which are not well addressed by existing short texts-specific topic modelling approaches. In this paper, we identify these issues and quantitatively and qualitatively evaluate how well a set of selected short texts-specific topic modelling approaches perform as IoT service categorisers against these issues. The results show that these approaches do not perform well in a corpus of noisy APIs descriptions and heterogeneous service descriptions. Also, they do not support domain identification of services, which is essential in domain-based service search. We conclude that integrating an appropriate and comprehensive knowledge base (i.e., domain ontology) could minimise noise and address IoT's APIs and service descriptions' heterogeneity. More importantly, it can identify the domains of those APIs and services.

**INDEX TERMS** Large-scale IoT, service discovery, service oriented computing, topic modelling.

## I. INTRODUCTION

By enabling easy access to, and interaction with a wide variety of physical devices or things, the IoT will foster the development of various applications in many different domains [1]. Although exciting, there are significant scientific and technological challenges to be overcome before these applications can be fully realised. These challenges arise, at least partially, due to the heterogeneity and scale of things/objects and their offered services [2]. The lack of semantic interoperability within IoT services presented in different semantic languages (i.e., WSDL-S, OWL-S) is an excellent example of the kinds of scientific problems that heterogeneity introduces. The adoption of service-oriented computing (SOC), especially large-scale SOC in IoT, can mitigate these challenges [3]–[7]. However, autonomous service search and discovery, which is an important aspect of SOC, in this large-scale, heterogeneous and multi-domain services is a challenging task for many reasons, including the

real-time requirements of many IoT applications and resource limitations of many IoT devices. Unified representation of heterogeneous IoT service descriptions and categorisation of these services into their application domains or clusters can reduce the search space and offer semantic interoperability. This will make service search [6]–[9] in IoT efficient (i.e., lower search time and consuming less resource) and scalable. Significantly, this will complement many existing service search approaches [10], [11], including a keyword-based search of things or services, by reducing the search space.

In many areas, including the autonomous categorisation of text, machine learning (ML) techniques are preferred over knowledge engineering (KE) [12], because of their effectiveness, considerable savings in terms of expert labour power and straightforward portability to different domains. Generally, ML approaches for categorisation fall into one of two schools of thought: discriminative or generative models. Recently, topic modelling (TM), which exploits a generative model (i.e., Probabilistic Latent Semantic Analysis, Latent Dirichlet Allocation [13]) or a combination of generative

The associate editor coordinating the review of this manuscript and approving it for publication was Jemal H. Abawajy<sup>1</sup>.

models (i.e., Hidden Topic Markov Model [14]), is being widely used in text categorisation, including news grouping, opinion mining, sentiment analysis of social networks' messages and IoT service descriptions, as they are statistically efficient (i.e., smaller training data), computationally efficient, and robust to missing values [9], [13], [15]–[17]. A categorisation of IoT services according to their domains, can provide scalability and efficient application-driven service search by reducing the search space [18]. Also, topic-based homogeneous and lower-dimensional representation, built automatically from the syntactic/semantic IoT service descriptions, can semantically search services regardless of their providers, description formats or technologies [16], [17]. However, research efforts [9], [17] is limited in this field, and they are based on Latent Dirichlet Allocation (LDA) [13], which does not perform well in short texts because of their sparseness. Recently, many TM-based short texts (e.g., tweets, or Facebook status) categorisation approaches [15], [16], [19]–[23], have proposed novel methods to improve the performance of topic-based categorisers in short texts. These proposals can be useful in IoT service descriptions categorisation as they are generally short and sparse. In this context, an evaluation of these proposals on heterogeneous IoT service descriptions will provide a more in-depth perspective of the state-of-the-art. There is no such work that evaluates these proposals.

Considering the importance of categorisation of IoT services, this paper: (1) identifies the issues of IoT services categorisation; (2) presents an evaluation of a set of existing TM-based short texts categorisation approaches, in the light of IoT services, to show how well they perform as categorisers and address the identified issues; and (3) outlines open research challenges, recommending future research directions. The evaluation results show that TM-based approaches can be used to categorise IoT services. However, few open issues, including distributed implementation, hierarchical categorisation, the order of words/concepts, and distributed knowledge base (i.e., domain ontology) development and management, still exist. However, research is needed, especially in domain-specific context or knowledge integration within the approaches, to support domain identification and improve these techniques' accuracy and precision. Also, research is needed in distributed implementation and distributed knowledge base (i.e., domain ontology) development and management.

Section II presents a list of issues in categorising IoT services and potential support for these issues from existing topic modelling approaches. Section III provides an overview of the selected approaches and justify their selection for further study. Section IV presents the framework, testbed, datasets, and metrics for the evaluation. Section V presents the results of the evaluation. Qualitative evaluation and open research challenges, including possible future research directions, are presented in Section VI. Section VII concludes the work and points to areas of potential future work.

**TABLE 1. List of abbreviations and acronyms.**

| Acronyms | Abbreviations                        |
|----------|--------------------------------------|
| IoT      | Internet of Things                   |
| SOC      | Service-oriented Computing           |
| TM       | Topic Modelling                      |
| ML       | Machine Learning                     |
| LDA      | Latent Dirichlet Allocation          |
| WS       | Web Services                         |
| TF-IDF   | Frequency-Inverse Document Frequency |
| SATM     | Self-Aggregation-based Topic Model   |
| TLDA     | Twitter-LDA                          |
| DMM      | Dirichlet Multinomial Mixture        |
| GSDMM    | Gibbs Sampling DMM                   |
| BTM      | Biterm Topic Model                   |
| LFDMM    | Latent features based DMM            |
| GPUDMM   | Generalised Poly urn DMM             |
| GPUPDMM  | Poisson-based GPUDMM                 |
| LFLDA    | Latent Feature LDA                   |
| GW       | Gateway                              |
| DS1      | First Dataset                        |
| DS2      | Second Dataset                       |
| DS3      | Third Dataset                        |
| DS4      | Fourth Dataset                       |
| H        | Homogeneity                          |
| C        | Completeness                         |
| NMI      | Normalized Mutual Information        |
| TC       | Time Complexity                      |

## II. IoT SERVICES CATEGORISATION: ISSUES

Unlike categorisation for other short texts, IoT services categorisation encounters a few specific issues. In the following, we identify these issues, and in later sections, we will use them to evaluate a set of existing short texts-specific TM-based categorisers.

### A. HETEROGENEITY

Different providers can offer the same service and similar services. Service providers may advertise or register their services through syntactic or semantic service descriptions written in different languages (e.g., JSON, XML, OWL-S, WSDL-S). Alternatively, they can register their services as Web Services (WS) or REST services through API descriptions. Such service representation-level heterogeneity may cause semantic mismatches and make search inefficient. Also, services' content-level heterogeneity (i.e., atomic- contains a domain or topic, composite- may contain multi-domain services) could make a single solution inefficient for a different type of content in services.

### B. SCALABILITY

The large-scale IoT services are likely to result in many heterogeneous concepts to describe those services. Handling many concepts in resource-constrained IoT gateways (e.g., Raspberry Pi) is a challenging task, especially in terms of scalability.

### C. DOMAIN HIERARCHY

The IoT can offer its services to a large number of applications in numerous domains and environments. These domains can have hierarchical sub-domains and may have similar

services (e.g., temperature service for weather domain and body temperature service for healthcare domain), making services categorisation difficult.

#### D. SHORTNESS AND SPARSENESS

Generally, IoT service descriptions are short in length and are consequently much more sparse in word co-occurrences. As a result, approaches such as the Term Frequency-Inverse Document Frequency (TF-IDF) measure does not work well. Moreover, the use of the Vector Space Model [15] to represent short service descriptions, and the sparse and high-dimensional representation vectors, will result in a waste of both memory and computation time. On the other hand, limited and insufficient contexts make it more difficult to identify the senses of ambiguous words or concepts in short service descriptions. Many existing approaches [15], [16], [24]–[28] address data sparsity and related issues (i.e., context) in short texts. These approaches are grouped and briefly presented in Section III.

#### E. ORDER OF WORD IN DESCRIPTIONS

It is vital to maintain the order of words, especially the order of words on input, output, precondition and effect (IOPE) features of IoT services, during categorisation to support functionality (e.g., inputs, outputs, processes and operations) based search. However, the bag-of-words concept, used in TM-based categorisation, does not maintain the order of words/concepts in service descriptions. It relies on frequencies of words from a dictionary.

#### F. KNOWLEDGE BASE FOR CATEGORISATION

Knowledge bases, such as ontologies (i.e., domain ontology), are essential to categorise services, especially to identify services' domains. A domain-specific ontology building and management is a challenging task, especially in distributed IoT environments.

#### G. DISTRIBUTED IMPLEMENTATION

The IoT and its offered services are distributed, and so are the IoT service registries. In distributed service registries, a categoriser's centralised implementation may not work well mainly because of communication overhead and higher response time.

#### H. RESOURCE-CONSTRAINTS

Resource-constrained IoT gateways (e.g., Raspberry Pi, smartphone) require categorisation approaches with lower time and space complexity. They also need to be energy efficient.

### III. OVERVIEW OF SELECTED APPROACHES

Recent research into topic modelling for short texts has resulted in a number of proposals, which fall into two broad categories: mixed-membership models and mixture models. These models can be further categorized based on their short texts-specific solutions. We introduce the class of approach,

briefly compare existing work and justify the selection of one approach for further quantitative evaluation.

*Aggregation of Short Texts:* Many mixed-membership proposals [22], [27], [29], [30] aggregate short texts based on metadata (e.g., location, timestamps, hashtags) to form a pseudo-document and then apply conventional topic modelling on it. However, favourable metadata may not always be available in IoT service descriptions. TwitterRank [29], an extension of the PageRank algorithm, is not suitable for IoT services categorisation, as it ranks (not categorises) users (not tweets or documents) based on their influence on Twitter. Also, the unavailability of hashtags/labels makes the hashtags-based aggregation dependent LDA [27] approach unsuitable for IoT services categorisation. The self-aggregation based topic model (SATM) [30] assumes that each short text is a segment of a long pseudo-document and shares the same topic proportion of the pseudo-document, which may not be accurate in IoT service descriptions. Setting an appropriate number of long pseudo-documents in SATM is a non-trivial task. Moreover, the inference process involving both text aggregation and topic sampling is time-consuming. Twitter-LDA (TLDA) [22] could be a potential TM-based categorisation approach for IoT services, assuming that each service description comes from a topic or domain, and all the services registered by a service provider in a service registry are from a domain or its sub-domains. This is a realistic assumption as generally a service provider, such as a healthcare service provider, offers and registers healthcare-related services, and so TLDA is further evaluated in this study.

*One Topic per Document/Service Description:* A simple and effective approach for short texts is to restrict the document-topic distribution by assuming only one topic per document. Given the limited content of short texts, this assumption is reasonable and is proven to be more effective than conventional topic models in many studies [15], [31]. The Dirichlet Multinomial Mixture (DMM) [31] model may not be suitable for resource-constrained IoT gateways because of its resource-hungry parameter estimation method (Gibbs Sampling). GSDMM [15] is an improved version of the DMM, especially for short texts using collapsed Gibbs Sampling (GS). The collapsed GS offers better performance, especially in terms of computing resources and time than the Gibbs Sampling because it examines uncertainty in smaller space. GSDMM is further evaluated in this study.

*Explicit Word Co-Occurrences Modelling:* Recently, few proposals, including [28] have worked towards intensifying the word co-occurrence information from the collection of short texts being modelled. One of the key reasons for the poor performance of the conventional topic models [13], [32] in short texts is their implicit modelling of document-level word co-occurrence. The biterm topic model (BTM) [28] explicitly models the generation of word co-occurrence patterns instead of single words, as in many topic models [13], [32]. Also, it exploits the aggregated patterns learned from the whole corpus for learning topics to solve the problem

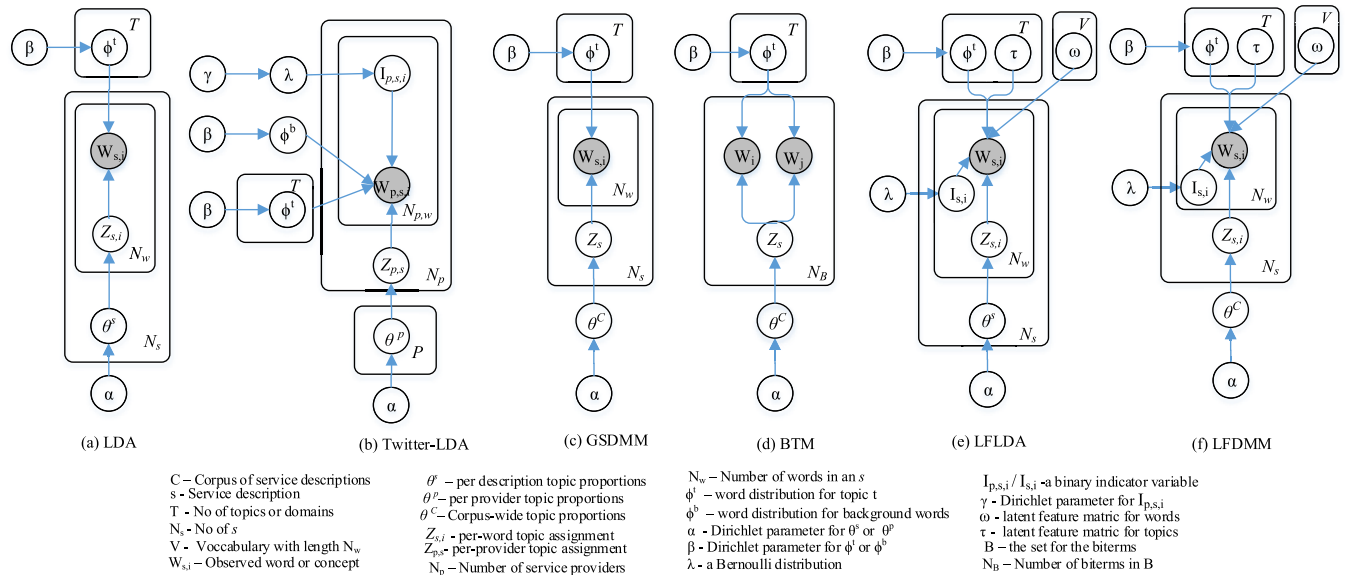


FIGURE 1. Selected TM Approaches.

of sparse word co-occurrence patterns at the document-level. As the aggregation process does not require external data or metadata, it could be useful for IoT services categorisation, and so it is further evaluated in this work.

**Words Embedding:** Integration of word embedding into TM approaches can add context to the service descriptions and improve their categorisation performance [16], [19], [33], [34], as they encode both syntactic and semantic information of words into continuous vectors in which similar words are close in vector space. Latent features based DMM (LFDMM) [19] and generalised Polya urn DMM (GPUDDMM) [16] are two recent proposals on words embedding. Poisson-based Dirichlet Multinomial Mixture model (GPUDDMM) [34] is an extended version of GPUDDMM, which allows each document can be generated by one or more (but not too many) topics. These proposals are similar in terms of their working principle (exploits word embedding approach to add context to service descriptions). We have selected LFDMM over GPUDDMM as it integrates word embedding into both LDA (mixed-membership model) and DMM/GSDMM (mixture model).

### A. SELECTED APPROACHES

The key components of a generative model, including a topic model are: (i) a generation process and (ii) a model parameters estimation method for the generation process. The generation processes of the selected approaches are illustrated in Figure 1 and described in Algorithms 1-6. The figure also includes all the necessary notations used in the algorithms. These approaches use one of the three popular model parameters estimation methods: (i) variational inference, (ii) Gibbs Sampling, and (iii) collapsed Gibbs Sampling. They are briefly presented in the following in terms of IoT service descriptions.

### Algorithm 1 Generation Process for LDA

- 1: Draw each topic  $\phi^t \sim Dir(\beta)$ ,  $t = 1, \dots, T$
- 2: **for** each service description  $s$  in  $S_H$ : **do**
- 3:   Draw topic proportions  $\theta^s \sim Dir(\alpha)$
- 4:   **for** each word  $i = 1, \dots, N_w$ : **do**
- 5:     Draw  $Z_{s,i} \sim Mult(\theta^s)$
- 6:     Draw  $W_{s,i} \sim Mult(\phi^{Z_{s,i}})$
- 7:   **end for**
- 8: **end for**

#### 1) LDA

LDA [13] represents each document or service description  $s$  as a probability distribution  $\theta^s$  over  $T$  topics or domains, where each topic  $z$  is modelled by a probability distribution  $\phi^z$  over words or concepts in the vocabulary set  $V$ . Figure 1 (a) illustrates, and Algorithm 1 describes the generation process of service descriptions for LDA. LDA [13] uses a variational inference model to infer the parameters used in the generation process.

#### 2) Twitter-LDA

Twitter-LDA or TLDA [22] is explicitly designed for tweets, which are short texts of only 140 characters. It considers that there are  $T$  topics in  $C$ , the corpus of tweets, each represented by a word distribution. For IoT service categorisation, we replace tweets with service descriptions. Unlike LDA, TLDA considers topic distribution over all the tweets generated by a user, which turns short messages into a longer pseudo-message. Also, TLDA includes a background model to generate a list of background words (e.g., yeah, and great). IoT services' background model can be used to generate words or concepts, including service, IoT, endpoint,

**Algorithm 2** Generation Process for Tiwtter-LDA

---

```

1: Draw  $\phi^b \sim Dir(\beta)$ ,  $\lambda \sim Dir(\gamma)$ 
2: Draw each topic  $\phi^t \sim Dir(\beta)$ ,  $t = 1, \dots, T$ 
3: for each service provider  $p = 1, 2, \dots, P$  : do
4:   Draw topic proportions  $\theta^p \sim Dir(\alpha)$ 
5:   for each service description  $s = 1, \dots, S_n$  do
6:     Draw  $Z_{p,s} \sim Mult(\theta^p)$ 
7:     for each word  $i = 1, \dots, N_{p,w}$  do
8:       Draw  $I_{p,s,i} \sim Mult(\pi)$ 
9:       if  $I_{p,s,i} = 0$  then
10:        Draw  $W_{p,s,i} \sim Mult(\phi^b)$ 
11:       else
12:        Draw  $W_{p,s,i} \sim Mult(\phi^{z_{p,s}})$ 
13:       end if
14:     end for
15:   end for
16: end for

```

---

for IoT service descriptions or API descriptions. Figure 1 (b) illustrates and Algorithm 2 describes the generation process of IoT service descriptions for TLDA.  $\lambda$ , a Bernoulli distribution, governs the choice between background concepts and domain/topic concepts. In writing a service description, a provider first chooses a topic based on its topic distribution for its domain, and it chooses a bag of concepts, one by one, based on the chosen topic or the background model. TLDA uses Gibbs sampling to infer the model parameters used in the generation process.

## 3) GSDMM

GSDMM [15] includes a clustering approach along with the collapsed Gibbs Sampling based parameters estimation algorithm. In the clustering approach, a document/service description chooses a cluster/domain with more documents/service descriptions and whose document/service descriptions share similar topic/topics. Following these rules, some clusters will grow larger, and others will vanish. The generation process of GSDMM is illustrated in Figure 1 (c) and described in Algorithm 3. As shown in Figure 1 (c), GSDMM uses the corpus-wide topic distribution  $\theta^C$  instead of LDA's document-wide topic distribution  $\theta^s$  and assigns a topic for a document/service description  $Z_s$  instead of LDA's word-wise topic assignment  $Z_{s,i}$  (Figure 1 (a)). These changes are to address the shortness and sparsity of short documents/service descriptions.

## 4) BTM

BTM [28] learns topics over short texts or service descriptions based on the aggregated biterms in the whole corpus to tackle the sparsity problem in a single document or service description. The direct modelling of biterm co-occurrence pattern, rather than a single word, offers semantic information of topics. It considers that the whole corpus as a mixture of topics ( $\theta^C$ ), where each biterm ( $b$ ) (e.g., mobile sensor,

**Algorithm 3** Generation Process for GSDMM

---

```

1: Draw for the corpus  $\theta^C \sim Dir(\alpha)$ 
2: Draw each topic  $\phi^t \sim Dir(\beta)$ ,  $t = 1, \dots, T$ 
3: for each service description ( $s$ ) in  $S_n$ : do
4:   Draw topic  $Z_s \sim Mult(\theta^C)$ 
5:   for each word  $i = 1, \dots, N_w$ : do
6:     Draw  $W_{s,i} \sim Mult(\phi^{z_s})$ 
7:   end for
8: end for

```

---

**Algorithm 4** Generation Process for BTM

---

```

1: Draw for the corpus  $\theta^C \sim Dir(\alpha)$ 
2: Draw each topic  $\phi^t \sim Dir(\beta)$ ,  $t = 1, \dots, T$ 
3: for each biterm  $b$  in the biterm set  $B$  : do
4:   Draw a topic assignment  $Z_b \sim Mult(\theta^C)$ 
5:   Draw two words:  $w_i, w_j \sim Mult(\phi^t)$ 
6: end for

```

---

body sensor) is drawn from a specific topic ( $Z_b$ ) independently. The probability that a biterm drawn from a specific topic is further captured by the chances that both words in the biterm are drawn from the topic ( $Z_b$ ). The generation process of BTM is illustrated in Figure 1 (d) and described in Algorithm 4. BTM uses Gibbs sampling to infer the model parameters used in the generation process. One prerequisite for BTM is the availability of biterms co-occurrence patterns in the corpus as insufficient patterns may deteriorate the performance of BTM instead of improving it.

## 5) LFLDA AND LFDMM

Latent Feature LDA (LFLDA) extends LDA and LFDMM [19] extend DMM/GSDMM for short texts by replacing their topic-to-concept  $\phi^t$  component that generates concepts/words from topics, with a two-component mixture of a topic-to-concept component  $\phi^t$  and a latent feature component  $\tau$ . Figure 1 (e) and (f)) have illustrated this extension along with the  $I_{s,i}$ , a Bernoulli distributed indicator function that determines whether the word/concept  $W_{s,i}$  is to be generated by the Dirichlet multinomial or latent feature component. Algorithms 5 and 6 describe the generation process of concepts/words in LFLDA and LFDMM. The larger corpora selection, which is relevant and comprehensive to extract features for service domains, is vital to shape the topic representations of short texts or service descriptions and improve the concept/word-topic mapping for them.

**IV. EXPERIMENTAL SETUP****A. THE EVALUATION FRAMEWORK**

Figure 2 presents the evaluation framework. We used it to evaluate the categorisation performance of the selected TM approaches. None of the selected approaches supports distributed implementation. Similar to [9], we used a hybrid implementation method to make these approaches suitable for IoT services. The implementation includes a centralised

**Algorithm 5** Generation Process for LFLDA

- 1: Draw each topic  $\phi^t \sim Dir(\beta), t = 1, \dots, T$
- 2: Get Vector representations for words  $\omega$
- 3: Get Vector representations for topics  $\tau$
- 4: Calculate latent feature matrix  $\tau\omega$
- 5: **for** each service description ( $s$ ) in  $S_n$ : **do**
- 6:     Draw topic proportions  $\theta^s \sim Dir(\alpha)$
- 7:     **for** each word  $i = 1, \dots, N_w$ : **do**
- 8:         Draw  $Z_{s,i} \sim Mult(\theta^s)$
- 9:         Draw  $I_{s_i} \sim Ber(\gamma)$
- 10:         Draw  $W_{s,i} \sim (1 - I_{s,i})Mult(\phi^{z_{s,i}}) + I_{s_i}Mult(\tau^{z_{s,i}}\omega^T)$
- 11:     **end for**
- 12: **end for**

**Algorithm 6** Generation Process for LFDMM

- 1: Draw for the corpus  $\theta^C \sim Dir(\alpha)$
- 2: Draw each topic  $\phi^t \sim Dir(\beta), t = 1, \dots, T$
- 3: Get Vector representations for words  $\omega$
- 4: Get Vector representations for topics  $\tau$
- 5: Calculate latent feature matrix  $\tau\omega$
- 6: **for** each service description ( $s$ ) in  $S_n$ : **do**
- 7:     Draw topic  $Z_s \sim Mult(\theta^C)$
- 8:     **for** each word  $i = 1, \dots, N_w$ : **do**
- 9:         Draw  $I_{s_i} \sim Ber(\gamma)$
- 10:         Draw  $W_{s,i} \sim (1 - I_{s,i})Mult(\phi^{z_s}) + I_{s_i}Mult(\tau^{z_s}\omega^T)$
- 11:     **end for**
- 12: **end for**

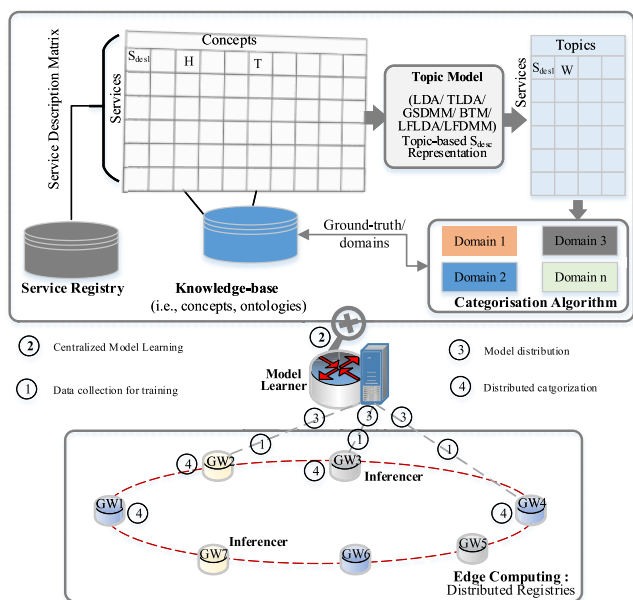
services, including an ontology, concepts, the ground truth of the services, (iii) a topic model and (iv) a categorisation algorithm. The centralised model learner and distributed inferencing nodes (i.e., IoT gateways) include all these components, but an inferencing node’s registry includes only locally registered services. The categorisation includes a four steps process as below.

- *Step 1:* The distributed IoT gateways (GW) send their registered service descriptions to the model learner, which accumulates the descriptions into a corpus for model learning. The accumulation process stops when the learner node has four or more service descriptions of each service category or domain. Categories that have less than four descriptions are not included in the training phase.
- *Step 2:* The model learner node runs a TM algorithm on the corpus and learns the model parameters (i.e.,  $\theta^s$ : a service description’s ( $s$ ) topic distribution).
- *Step 3:* The model learner broadcasts/multicasts the learned model and learned vocabulary to the inferencing nodes.
- *Step 4:* Based on the received model, each inferencing node runs its categorisation algorithm on the locally registered services and assign their topics or domains. The categorisation algorithm assigns a cluster to a service description according to GSDMM’s clustering algorithm, presented in Section III. This is because of its superior performance over the popular K-means algorithm [15]. Every inferencing node records the percentage of new concepts that appears from the services that registered after the first learned model. If the percentage reaches a threshold (e.g., 20% words in a sentence are new), the node sends those service descriptions to the model learner. The model learner relearns the model if it receives such service descriptions from one or more inferencing nodes.

**B. THE TESTBED**

We implemented the framework using two configurations: (i) the model learner on a Desktop with Intel i7-4770 3.4GHz 64-bit CPU and 8GB memory, and inferencing nodes (IoT GWs) are on Raspberry Pi 4 [35] with a Quad core Cortex-A72 (ARM v8) 64-bit 1.5GHz CPU and 4GB memory [35], (ii) the model learner and inferencing nodes (IoT GWs) on Raspberry Pi 4 [35]. The second configuration is to evaluate the approaches in resource-constrained IoT environments.

Figure 3 presents the physical deployment of the second implementation of the framework, which was a part of our IoT services implementation testbed [36]. Here, we briefly present the testbed implementation, and for detailed information of it, readers are referred to the work [36]. We used five Raspberry Pi 4 as gateways connected in a WiFi MANET. All these gateways installed distributed service registry, and four of them (Gateways 1-4) worked as inferencing nodes, and the fifth (unmarked) gateway (Figure 3) worked as the model learner. Two Raspberry Pi 4, three Galileo boards,



**FIGURE 2.** The evaluation framework.

model learning phase and a distributed inference or categorisation phase.

The key components of the framework are: (i) a service registry with accumulated service descriptions, (ii) a knowledge-base to store knowledge about application domains of the

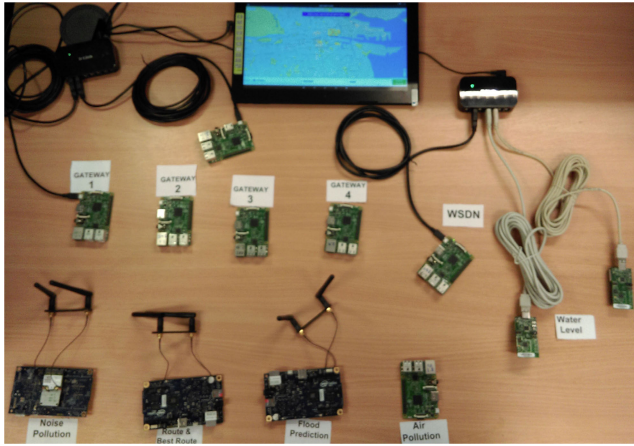


FIGURE 3. The physical deployment.

and six motes were used as service providers for dataset 1 (discussed in IV-C). They also used WiFi to communicate with gateways. On the other hand, five gateways worked as the inferring nodes in the first implementation and were connected with the model learner (the desktop) through WiFi.

### C. DATASETS

Generally, registered IoT service representations can take one of the three forms: (i) service descriptions written in different languages (i.e., JSON, XML, OWL-S, WSDL-S), (ii) Web Services (WS) written in different languages (i.e., OWL-S, WSDL-S), and (iii) REST API descriptions. The experiment considers four different datasets for training the models: one dataset for each of the three service representations forms and one for combining these three. The first dataset (DS1) [36] consists of 80 IoT service descriptions defined in JSON. These services are from four different domains. The second dataset (DS2) has been scraped from the ProgrammableWeb API directory [37]. The DS2 consists of 233 IoT services' API descriptions, which come from different domains (approximately 19). Unlike DS1, the DS2 is a noisy dataset that includes many background words, and several domains have only a few samples (e.g., the agriculture domain has only four descriptions). The third dataset (DS3) consists of 310 web service descriptions and is obtained from the OWL-S service retrieval test collection called OWLSTC v3 [38]. These services can be grouped into ten different domains. The fourth and final dataset (DS4) is a combination of the earlier three datasets and consists of 623 service descriptions from 31 different domains. DS4 was created to study the performance of the selected approaches in heterogeneous service representations (i.e., JSON for DS1, OWL-S for DS2, API descriptions). Table 2 summarises the datasets, where sparsity is defined as the % of nonzero elements in a vector representation of a service description.

Each of the above training datasets (DS1-DS4) has a representative inference or test dataset (DS1I-DS4I) to study the categorisers' performance in the new dataset through

TABLE 2. Summary of the datasets.

| Features       | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 |
|----------------|-----------|-----------|-----------|-----------|
| Avg. Length    | 17        | 43        | 12        | 24        |
| Min. Length    | 13        | 11        | 4         | 4         |
| Max. Length    | 20        | 115       | 30        | 115       |
| Voca. Size     | 67        | 2237      | 562       | 2676      |
| Avg. Sparsity  | 25.3%     | 1.9%      | 2.2%      | .89%      |
| Corpus length  | 80        | 233       | 310       | 623       |
| No. of domains | 4         | 19        | 10        | 31        |

topic inferring. Every dataset in DS1I-DS4I includes ten new service descriptions.

### D. EVALUATION METRICS

We introduce evaluation metrics for the categorisers and their impact on service search separately. The metrics to evaluate the categorisers are further divided into quantitative and qualitative ones. The quantitative metrics are Purity, Homogeneity (H), Completeness (C), Normalized Mutual Information (NMI) and time complexity (TC). These metrics are used in many existing work [15], [16], [28] to evaluate TM-based categorisers. In the following, we present the definitions of these metrics.

#### 1) PURITY

Purity measures the percent of the total number of services that were categorised or clustered correctly.

$$purity(C, GT) = \frac{1}{N} \sum_k \max_j |c_k \cap gt_j| \quad (1)$$

where  $C = c_1, c_2, \dots, c_k$  is the set of clusters and  $GT = gt_1, gt_2, \dots, gt_j$  is the set of ground truth (GT) or classes.

#### 2) HOMOGENEITY (H)

Homogeneity represents the percentage/fraction of services that came from a single category/cluster.

$$H(C, GT) = 1 - \frac{E(GT|C)}{E(GT)} \quad (2)$$

where  $E(GT|C)$  is the conditional entropy of the ground truth/classes given the cluster assignments and  $E(GT)$  is the entropy of the ground truth/classes.

#### 3) COMPLETENESS (C)

It represents the percentage/fraction of a given category/cluster of services are assigned to the same category/cluster.

$$C(C, GT) = 1 - \frac{E(C|GT)}{E(C)} \quad (3)$$

where  $E(C|GT)$  is the conditional entropy of clusters given the ground truth/classes and  $E(C)$  is the entropy of clusters.

#### 4) NORMALIZED MUTUAL INFORMATION (NMI)

Is a function that measures the agreement of the two assignments, ignoring permutations [39].

$$NMI(C, GT) = \frac{I(C, GT)}{[E(C) + E(GT)]/2} \quad (4)$$

**TABLE 3. Training performance of the categorisers.**

| Metrics              | Dataset | LDA  | TLDA | GSDMM | BTM  | LFLDA | LFDMM |
|----------------------|---------|------|------|-------|------|-------|-------|
| Purity               | DS1     | .97  | 1.0  | 1.0   | .72  | .75   | .75   |
|                      | DS2     | .44  | .46  | .465  | .37  | .36   | .38   |
|                      | DS3     | .74  | .80  | .78   | .54  | .37   | .44   |
|                      | DS4     | .54  | .66  | .67   | .36  | .45   | .52   |
| NMI                  | DS1     | .92  | 1.0  | 1.0   | .63  | .86   | .83   |
|                      | DS2     | .34  | .35  | .35   | .26  | .25   | .26   |
|                      | DS3     | .63  | .79  | .78   | .46  | .21   | .43   |
|                      | DS4     | .60  | .68  | .67   | .45  | .47   | .52   |
| H                    | DS1     | .928 | 1.0  | 1.0   | 1.0  | 1.0   | 1.0   |
|                      | DS2     | .30  | .305 | .32   | .25  | .23   | .425  |
|                      | DS3     | .715 | .791 | .778  | .46  | .22   | .24   |
|                      | DS4     | .615 | .65  | .646  | .48  | .47   | .5    |
| C                    | DS1     | .928 | 1.0  | 1.0   | 1.0  | 1.0   | 1.0   |
|                      | DS2     | .31  | .31  | .32   | .25  | .23   | .42   |
|                      | DS3     | .72  | .79  | .78   | .46  | .22   | .25   |
|                      | DS4     | .61  | .65  | .64   | .49  | .47   | .5    |
| Time complexity (c1) | DS1     | .11  | .10  | .035  | .05  | .22   | .21   |
|                      | DS2     | .19  | .18  | .12   | .13  | 10.1  | 10.2  |
|                      | DS3     | .16  | .14  | .07   | .08  | .571  | .572  |
|                      | DS4     | .21  | .22  | .19   | .178 | 20.2  | 20.7  |
| Time complexity (c2) | DS1     | .13  | .12  | .05   | .08  | .41   | .42   |
|                      | DS2     | .22  | .21  | .15   | .175 | 19.7  | 19.9  |
|                      | DS3     | .19  | .18  | .08   | .095 | 1.75  | 1.8   |
|                      | DS4     | .25  | .27  | .24   | .29  | 39.5  | 40.1  |

where  $I(C, GT)$  is the mutual information between  $C$  and  $GT$ ,  $E(C)$  is the entropy of  $C$  and  $E(GT)$  is the entropy of  $GT$ .

##### 5) TIME COMPLEXITY

It represents the time required by an approach to learn its model.

The qualitative metrics are derived from the issues identified in Section II are heterogeneity, hierarchical categorisation, scalability, order of words, support for ontology complexity and implementation. We also consider domain identification (supervised learning or classification) as a qualitative parameter. We use search response time and scalability to study the impact of service categorisation on service search.

## V. RESULTS

This section presents the quantitative evaluation results of the categorisers in terms of purity, NMI, H, C and time complexity, and the influence of the topics/domains number (T) and models' hyperparameters  $\alpha$  and  $\beta$  on categorisation purity. It also discusses how well the selected approaches address the issues identified in Section II. Finally, it demonstrates the potential of service categorisation in search.

### A. COMPARISON OF CATEGORISERS

Table 3 illustrates the results for purity, NMI, H, C and time complexity on training datasets (DS1-DS4), and Table 4 illustrates the results for purity on the corresponding inference datasets DS1I-DS4I. Like [15], [16], [28], we used symmetric values for  $\alpha$  and  $\beta$ , and they are fixed to  $\alpha = .25$  and  $\beta = .01$  for all the datasets. All the approaches perform well (e.g.,  $\text{purity} \geq .72$ ) in DS1 and DS1I as the service descriptions for DS1 and DS1I are structured, and they are short but

**TABLE 4. Inferring performance of the categorisers.**

| Metrics | Dataset | LDA  | TLDA | GSDMM | BTM  | LFLDA | LFDMM |
|---------|---------|------|------|-------|------|-------|-------|
| Purity  | DS1I    | .925 | 1.0  | 1.0   | .47  | .72   | .72   |
|         | DS2I    | .42  | .435 | .44   | .34  | .34   | .35   |
|         | DS3I    | .72  | .78  | .765  | .525 | .357  | .425  |
|         | DS4I    | .52  | .65  | .66   | .345 | .44   | .50   |

not sparse (as shown in Table 2, on average, more than 25% elements of every service description vector are nonzero). In DS2 and DS2I, all the categorisers perform poorly, and the main reasons for this are: (i) unstructured and noisy descriptions (i.e., too many common words such as API, service), (ii) heterogeneous length of service descriptions, and (iii) sparsity of the descriptions. On the other hand, all the approaches perform better in DS3 and DS3I than DS2 and DS2I, as descriptions in DS3 and DS3I are more structured and less noisy and less heterogeneous in length than DS2 and DS2I. In DS4 and DS4I, the performances other than the time complexity of all the approaches have improved compared to those in DS2 and DS2I, as the DS4 and DS4I are less noisy and more structure than DS2 and DS2I. However, the performances are not as good as in DS1, DS1I, DS3 and DS3I. They include DS2 or service descriptions similar to DS2, and service descriptions' length level heterogeneity and sparsity are more than other datasets.

TLDA and GSDMM outperform other approaches in all training and inference/test datasets. Their performance's main reason is their strict assumption: one topic per service description and this assumption is valid for the datasets as most of their services came from one of the listed domains. LDA, the baseline TM, performs well in DS1 and DS1I but struggles in other datasets mainly because of sparsity. On the other hand, BTM does not perform well because there are



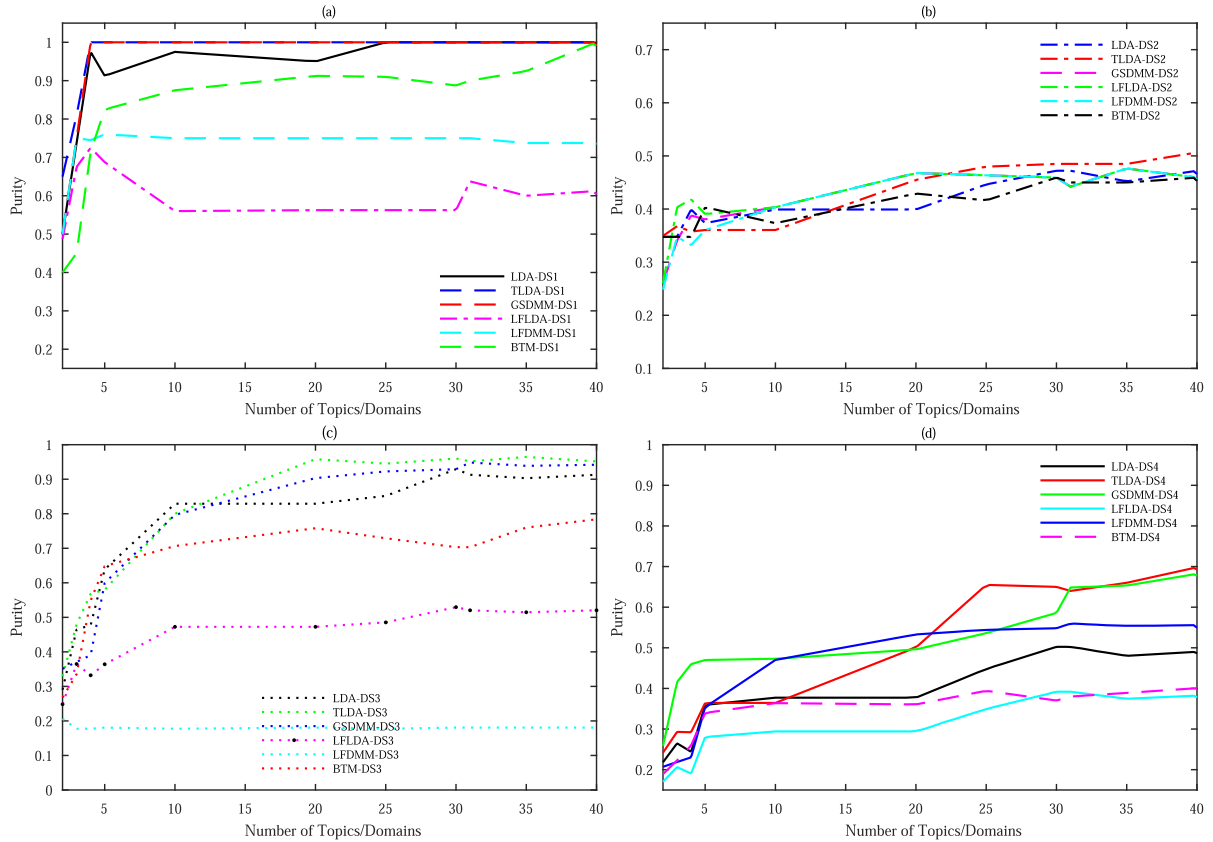


FIGURE 4. Number of topics/domains vs. clustering purity.

no or limited biterns co-occurrence patterns in the datasets. LFLDA and LFDMM perform poorly compared to TLDA, GSDMM and LDA. The reason for their poor performance is the inappropriate and irrelevant supporting corpus used in the word embedding. We used DS4 as a supporting corpus for all datasets to embed related words in service descriptions, and DS4 has many irrelevant words and topics, which have “used up” the vocabulary and topic space of LFLDA and LFDMM. The word embedding model of LFLDA and LFDMM represents each word using a single vector, which makes the model indiscriminative for ubiquitous homonymy and polysemy.

The purity results of all the approaches on the inference datasets show that these approaches are performing close to their training performance. Inference datasets with more differences than the training datasets may initiate retraining of the models to keep the models’ performance close to a certain threshold.

### 1) INFLUENCE OF NUMBER OF DOMAINS OR TOPICS (T)

In this section, we investigate the influence of  $T$  on the performance of clustering purity. An optimal number of  $T$  is important, as, for numbers below the optimal one, a categoriser may perform poorly. For numbers higher than the optimal one, a categoriser may need more computing resources to categorise the services. For example, as shown in Table 3,

the time complexity of an approach increases, not only by an increase in dataset size but also by the increase in the number of topics it involves. For this experiment, we use  $\alpha = .25$  and  $\beta = .01$  for all the datasets, and varied  $T$  from 2-40. As shown in Figure 4 (a-d), in all datasets, clustering purity improves as  $T$  increases till it reaches the optimal or near the actual number of  $T$ . After that, purity does not improve or deteriorate much. The reason for this pattern is the “richer gets richer” property of the clustering approach. For example, in DS1 (Figure 4 (a)), the actual number of  $T$  is 4, and all the approaches show the highest or close to the highest purity on this value. The purity values for the higher values of  $T$  do not fluctuate much.

### 2) INFLUENCE OF ALPHA $\alpha$

Dirichlet prior  $\alpha$  is a prior on the topic-distribution of documents (of a corpus), and it is a corpus-level parameter. It represents the sparsity of a document or service descriptions in terms of topic distribution (i.e., a lower value  $\alpha$  for corpus means every document in the corpus includes a few topics or one topic, not all the topics available in the corpus). Determining the optimal value or range of values of  $\alpha$  is important to represent the sparsity (topic-wise) of a document appropriately. For this experiment, we used  $\beta = .01$  for all the datasets,  $T = 4$  for DS1,  $T = 19$  for DS2,  $T = 10$  for

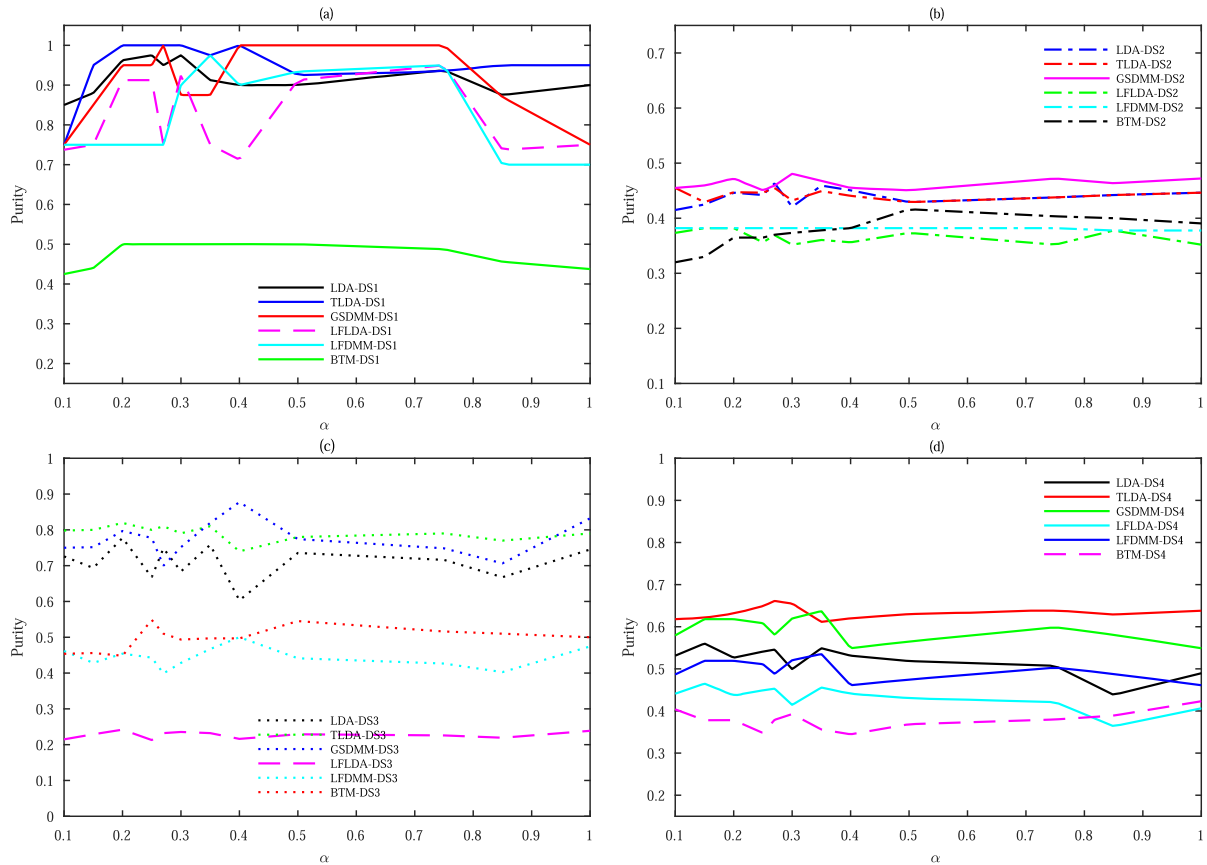


FIGURE 5. Sparsity of topic distribution ( $\alpha$ ) vs. clustering purity.

DS3 and  $T = 4$  for DS4, and varied  $\alpha$  from .1-1.0 (as service descriptions are short and sparse). As shown in Figure 5 (a-d), clustering purity is not the same for all values of  $\alpha$ . All the approaches' best performances may coincide at a range rather than a fixed value of  $\alpha$ . For example, in DS1 (Figure 5 (a)), the best performances of the categorisers coincide in between .25-.37. More importantly, as shown in Figure 5 (a-d), a fixed value or range of values may not work well in all datasets of service descriptions, as it is a property of a corpus instead of the property of a set of corpora. In short and heterogeneous datasets, the use of an adaptive and optimised value of  $\alpha$  based on a corpus may show better performance than a predefined value (e.g.,  $\alpha = 50/T$  [13]).

### 3) INFLUENCE OF BETA $\beta$

Dirichlet prior  $\beta$  is a prior on the word-distribution of topics (of a document/service description), and generally, it is a corpus-level (symmetric value) parameter. It represents the sparsity of a document or service descriptions in terms of word distribution (i.e., a lower value  $\beta$  represents each document in a corpus includes a few words, not all the words available in the corpus or corresponding vocabulary). Like  $\alpha$ , the optimal value or range of values of  $\beta$  is necessary to appropriately represent the sparsity (word-wise/feature-wise) of a document. For this experiment, we used  $\alpha = .25$  for all

the datasets,  $T = 4$  for DS1,  $T = 19$  for DS2,  $T = 10$  for DS3 and  $T = 4$  for DS4, and varied  $\beta$  from .01-.2 (as service descriptions are short and sparse). As shown in Figure 6 (a-d), clustering purity is not the same for all the values of  $\beta$ , and most approaches in all the datasets showed their optimal clustering purity within the range  $\beta = .01 - .07$ , which includes the recommended value of  $\beta$  (.01). The variation of clustering performance varies less with  $\beta$  than  $\alpha$ . The clustering performances of LFLDA and LFDMM vary more with  $\beta$  than that of LDA, TLDA, GSDMM and BTM. Explicit use of latent feature/concepts in LFLDA and LFDMM could be a potential reason for this behaviour. Moreover, BTM relies more on corpus-wide word co-occurrence than  $\beta$ . On the other hand, GSDMM assigns a topic to a document, and TLDA assigns topic proportions to a user or service provider.

## B. EFFICIENT AND SCALABLE SERVICE SEARCH

We implemented a service search use case in MongoDB [40]. In MongoDB, we created four different database collections (CL1-CL4) for four datasets DS1-DS4 (without categorisation) and a collection (CLD1-CLD31) for every categorised domain. We searched ten different services in CL1-CL4 and CLD1-CLD31 through a key-value pair based query and recorded their worst response time in milliseconds (ms). Figure 8 presents results. As shown in the figure, service

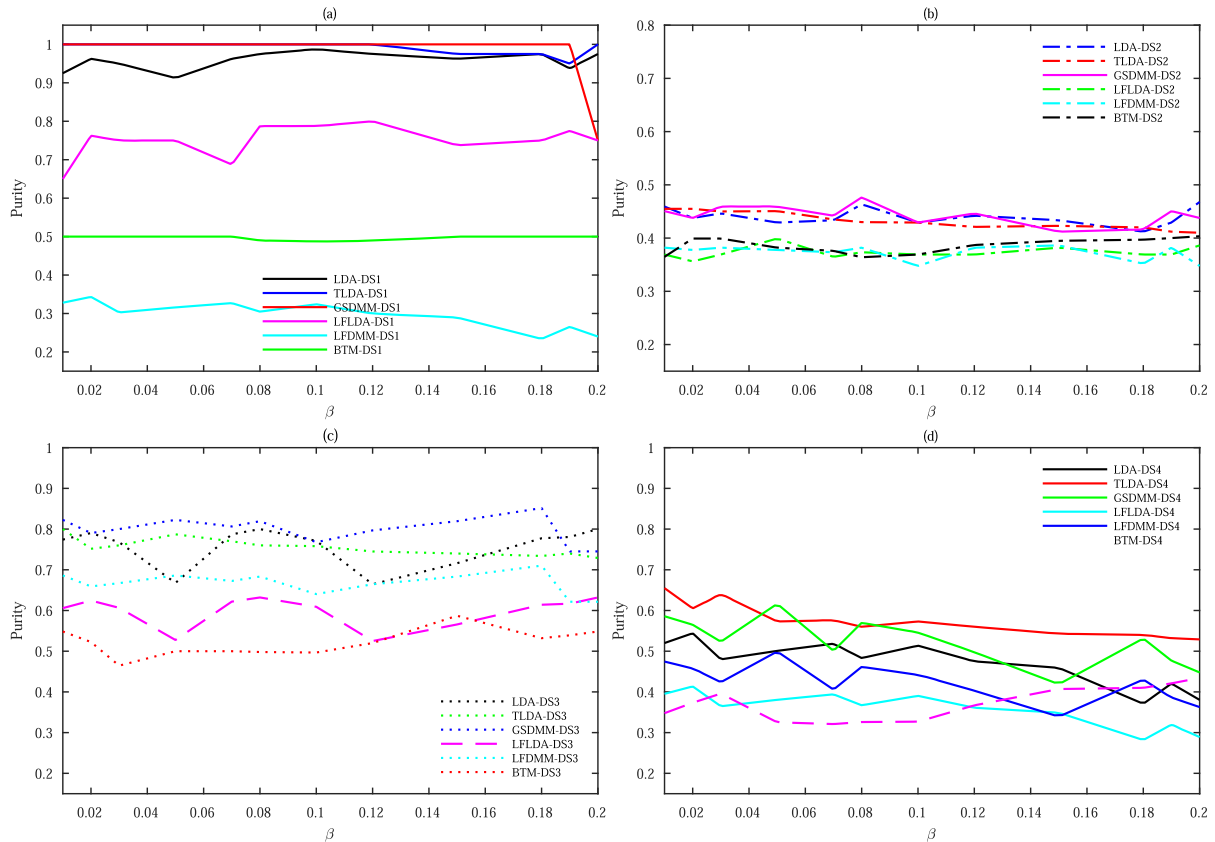


FIGURE 6. Sparsity of word distribution ( $\beta$ ) vs. clustering purity.

search is categorised, and reduced search space or collections scales better than the service search in non-categorised and larger search space. For example, the worst service search time in the non-categorised CLD2 is 2.5 ms and in the categorised CLD2 is 1 ms. Moreover, this lower response time will offer efficient service search in resource-constrained IoT devices by utilising computing resources and consuming battery power for a shorter time. Even though the worst search time for all the categorised collections in this experiment is 1 ms, it will increase if per cluster service descriptions count increases.

VI. QUALITATIVE EVALUATION AND OPEN RESEARCH CHALLENGES

Unlike discriminative models [18], TM-based categorisers can offer additional benefits by addressing the issues identified in Section II. In the following, we discuss how these issues can be addressed by existing TM-based approaches, including the selected ones. Although the selected approaches presented herein address many issues and requirements (Figure 7) in IoT service categorisation, there are still some open research challenges, which are also discussed in this section.

A. HETEROGENEITY

As illustrated in Section V, TM-based approaches can address service representation-level heterogeneity through

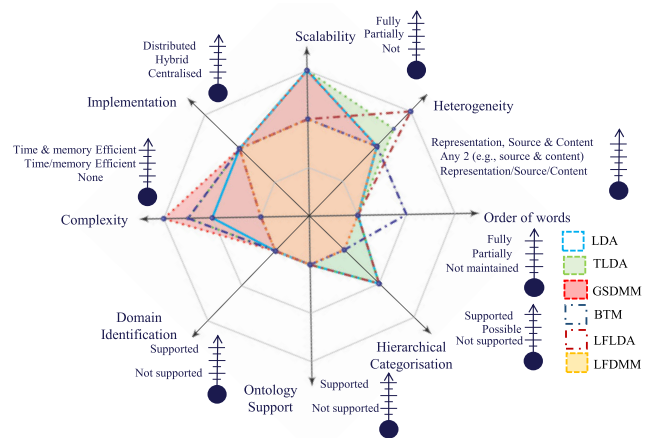


FIGURE 7. Qualitative evaluation.

topic-based unified service representation and categorise services according to their topic/topics. All the selected approaches address two types (i.e., representation and source, representation and content) of heterogeneity (Figure 7). Only TLDA and LFLDA [13], [22] address three types of heterogeneity because of their mixed membership models. However, LDA, a mixed membership model, may not always address source-level heterogeneity as it is not designed for short texts. Also, service descriptions' length (i.e., word length) level heterogeneity is not addressed well by the

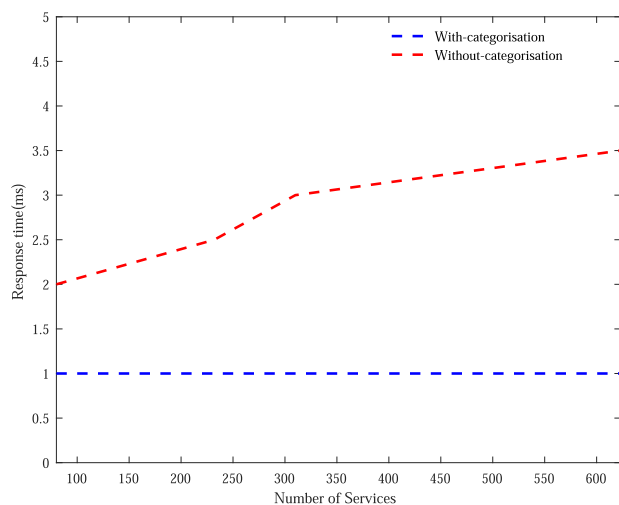


FIGURE 8. Scalable service search.

selected approaches. The use of adaptive and optimised hyperparameters selection in a TM may address this.

### B. SCALABILITY

Probabilistic topic models are popular methods for dimensionality reduction of text documents or images. For example, a topic-based representation of a service description from DS4 reduces the length (2676) of the word vector representation of the service description to 31. This lower-dimensional representation and topic-based categorisation of services can offer an efficient and scalable service search (Figure 8). As shown in Figure 7, LDA, TLDA, and GSDMM are more scalable than BTM, LFLDA and LFDMM as BTM needs corpus-wide biterm co-occurrence patterns, and LFLDA and LFDMM need a relevant and larger corpus, which may not always be available.

### C. HIERARCHICAL CATEGORISATION

The selected approaches do not support hierarchical categorisation. However, in mixed membership model-based approaches (i.e., LDA, TLDA and LFLDA), a service description includes multiple topics, which may be useful in hierarchical categorisation (Figure 7). On the contrary, strict condition about a topic (one topic per document or service description) of mixture model-based categorisers (e.g., GSDMM, GPUDMM) makes them unsuitable in hierarchical categorisation documents/services. An adaptive and less strict condition might allow more than one topics in a document/service description and support hierarchical categorisation. Further research is necessary for this direction.

### D. ORDER OF WORDS/CONCEPTS

As shown in Figure 7, only BTM partially maintains the order of words, explicitly exploiting biterm (e.g., body sensor, wireless network) co-occurrence patterns in a service description. However, the use of bi-grams or n-grams [41] based TM can maintain the order of words or phrases. The use of bi-grams

or n-grams may maintain the order of words/concepts in service descriptions, but it also increases complexity, especially in short texts, by making them shorter. For example, 3-grams words, such as body-sensor-network instead of 3 uni-gram words body, sensor and network, shorten the service description by two words. Context-aware adaptive use of n-grams in short texts could be a potential research direction.

### E. SUPPORT FOR KNOWLEDGE BASE/ ONTOLOGY

The selected approaches do not offer any support for ontology development and management. Also, word embedding based approaches (i.e., LFLDA, LFDMM) perform poorly due to the lack of distributed and comprehensive knowledge base (e.g., domain ontology, appropriate supporting and larger corpus) and the word embedding model's single vector assumption. Domain knowledge and TM-based categoriser may incrementally help each other to build and maintain a distributed knowledge base. This distributed knowledge base can also work as a vocabulary for supporting and larger corpus in word embedding based approaches. Also, context-aware word embedding [42] can address the homonymy and polysemy issue of single vector assumption. TM-based ontology [43] and taxonomies [44] learning, along with an overlay network of the service registries, could be a potential solution to distributed ontology development and maintenance.

### F. IMPLEMENTATION

The selected approaches do not support distributed implementation, but they support a hybrid implementation, as shown in Section IV. Distributed implementation of a topic model requires the parallel implementation of the model parameters estimation method. Generally, Gibbs Sampling is parallelizable, but efficient collapsed Gibbs Sampling is not. A trade-off is necessary between collapsed Gibbs sampling's efficiency and parallelizability.

### G. COMPLEXITY

As shown in Figure 7, LFLDA and LFDMM need more processing time and memory (complex) space compared to LDA, TLDA, GSDMM and BTM. Distributed implementation [45], [46], and model parameters estimation algorithm with lower time and space complexity [15] may reduce the complexity of these approaches.

### H. DOMAIN IDENTIFICATION

The selected approaches are clustering-based unsupervised categorisers and do not support domain identification. Use of feature selector to select appropriate domain features [47] or classifiers, such as SVM [16], [48] and K-Nearest Neighbor with topic modelling, can identify service domains. A domain-based IoT services classifier could be a potential future research direction.

## VII. CONCLUSION AND FUTURE WORK

In categorisation, unlike other short texts, IoT services encounter a few specific issues, which can be addressed by

topic modelling based categorisers. This article identified those issues and used them to evaluate six selected topic modelling based categorisers, namely LDA, TLDA, GSDMM, BTM, LFLDA and LFDMM in IoT services descriptions. All except LDA, the baseline model, are designed for short texts. The quantitative evaluation results show that all the approaches perform well in a short but not too sparse dataset (DS1). However, they do not perform well in noisy API descriptions (DS2) datasets and heterogeneous service descriptions (DS3 and DS4). TLDA and GSDMM outperform other approaches in all training and inference datasets mainly because of their strict assumption: one topic per service description. Word embedding based approaches (LFLDA and LFDMM) may support semantic interoperability, but they perform poorly compared to TLDA, GSDMM and LDA. They perform poorly because of the inappropriate supporting corpus.

In addition to the comparison of the categorisers, this article presents results on the influence of models' hyperparameters  $\alpha$  and  $\beta$ , and number of topics/domains (T) on categorisation purity. These results demonstrate that an optimal value of T can offer categorisation efficiency through lower processing time, and optimal values of  $\alpha$  and  $\beta$  can optimise categorisers' performance by well representing the datasets.

The evaluation of the selected approaches presented in this study shows that these approaches well address heterogeneity. They also have the potential to address scalability issues in IoT service categorisation. However, few open issues, including distributed implementation, hierarchical categorisation, the order of words/concepts, and distributed knowledge base (i.e., domain ontology) development and management, still exist. There is significant scope for future work in these areas. Realising the importance of domain identification in domain-based service search in IoT, our future endeavours will focus on developing and managing a distributed knowledge base (i.e., domain ontology), exploiting topic modelling's generative aspect. Our future effort will also be to evaluate our solutions in larger IoT datasets as the datasets used in this evaluation are insufficient to demonstrate service search's scalability.

## ACKNOWLEDGMENT

The author would like to thank Siobhán Clarke for her insightful suggestions and critical review of the manuscript.

## REFERENCES

- [1] A. Zanello, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.
- [2] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 70–95, Feb. 2016.
- [3] T. Teixeira, S. Hachem, V. Issarny, and N. Georgantas, "Service oriented middleware for the Internet of Things: A perspective," in *Proc. 4th Eur. Conf. Towards Service-Based Internet*. Berlin, Germany: Springer, 2011, pp. 220–229.
- [4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.
- [5] H. Panetto and J. Cecil, "Information systems for enterprise integration, interoperability and networking: Theory and applications," *Enterprise Inf. Syst.*, vol. 7, no. 1, pp. 1–6, Feb. 2013.
- [6] C. Cabrera, G. White, A. Palade, and S. Clarke, "The right service at the right place: A service model for smart cities," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2018, pp. 1–10.
- [7] C. Cabrera, A. Palade, G. White, and S. Clarke, "An urban-driven service request management model," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2020, pp. 1–7.
- [8] S. De, P. Barnaghi, M. Bauer, and S. Meissner, "Service modelling for the Internet of Things," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Sep. 2011, pp. 949–955.
- [9] G. Cassar, P. Barnaghi, W. Wang, and K. Moessner, "A hybrid semantic matchmaker for IoT services," in *Proc. IEEE Int. Conf. Green Comput. Commun.*, Nov. 2012, pp. 210–216.
- [10] Y. Qin, Q. Z. Sheng, N. J. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A survey on data-centric Internet of Things," *J. Netw. Comput. Appl.*, vol. 64, pp. 137–153, 2016.
- [11] C. Perera and A. V. Vasilakos, "A knowledge-based resource discovery for Internet of Things," *Knowl.-Based Syst.*, vol. 109, pp. 122–136, Oct. 2016.
- [12] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002.
- [13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
- [14] A. Gruber, M. Rosen-zvi, and Y. Weiss, "Hidden topic Markov models," in *Proc. Artif. Intell. Statist.*, 2007, pp. 163–170.
- [15] J. Yin and J. Wang, "A Dirichlet multinomial mixture model-based approach for short text clustering," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 233–242.
- [16] C. Li, H. Wang, Z. Zhang, A. Sun, and Z. Ma, "Topic modeling for short texts with auxiliary word embeddings," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2016, pp. 165–174.
- [17] G. Cassar, P. Barnaghi, and K. Moessner, "Probabilistic matchmaking methods for automated service discovery," *IEEE Trans. Services Comput.*, vol. 7, no. 4, pp. 654–666, Oct. 2014.
- [18] G. Pirrò, D. Talia, and P. Trunfio, "A DHT-based semantic overlay network for service discovery," *Future Gener. Comput. Syst.*, vol. 28, no. 4, pp. 689–707, Apr. 2012.
- [19] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, "Improving topic models with latent feature word representations," *Trans. Assoc. Comput. Linguistics*, vol. 3, pp. 299–313, Dec. 2015.
- [20] T. Lin, W. Tian, Q. Mei, and H. Cheng, "The dual-sparse topic model: Mining focused topics and focused terms in short text," in *Proc. 23rd Int. Conf. World Wide Web (WWW)*, 2014, pp. 539–550.
- [21] F. Wang, Z. Wang, Z. Li, and J.-R. Wen, "Concept-based short text classification and ranking," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, Nov. 2014, pp. 1069–1078.
- [22] W. X. Zhao, J. Jiang, J. Weng, J. He, E. P. Lim, H. Yan, and X. Li, "Comparing Twitter and traditional media using topic models," in *Proc. 33rd Eur. Conf. Adv. Inf. Retr.* Berlin, Germany: Springer, 2011, pp. 338–349.
- [23] R. Albalawi, T. H. Yeap, and M. Benyoucef, "Using topic modeling methods for short-text data: A comparative analysis," *Frontiers Artif. Intell.*, vol. 3, p. 42, Jul. 2020. [Online]. Available: <https://www.frontiersin.org/article/10.3389/frai.2020.00042>
- [24] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & Web with hidden topics from large-scale data collections," in *Proc. 17th Int. Conf. World Wide Web (WWW)*, 2008, pp. 91–100.
- [25] O. Jin, N. N. Liu, K. Zhao, Y. Yu, and Q. Yang, "Transferring topical knowledge from auxiliary long texts for short text clustering," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2011, pp. 775–784.
- [26] L. Hong and B. D. Davison, "Empirical study of topic modeling in Twitter," in *Proc. 1st Workshop Social Media Anal. (SOMA)*, 2010, pp. 80–88.
- [27] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie, "Improving LDA topic models for microblogs via tweet pooling and automatic labeling," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2013, pp. 889–892.
- [28] X. Cheng, X. Yan, Y. Lan, and J. Guo, "BTM: Topic modeling over short texts," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 2928–2941, Dec. 2014.
- [29] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "TwitterRank: Finding topic-sensitive influential twitterers," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining (WSDM)*, 2010, pp. 261–270.

- [30] X. Quan, C. Kit, Y. Ge, and S. J. Pan, "Short and sparse text topic modeling via self-aggregation," in *Proc. 24th Int. Conf. Artif. Intell. (IJCAI)*, 2015, pp. 2270–2276. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2832415.2832564>
- [31] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, nos. 2–3, pp. 103–134, 2000.
- [32] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, vol. 1, 2009, pp. 248–256.
- [33] R. Das, M. Zaheer, and C. Dyer, "Gaussian lda for topic models with word embeddings," in *Proc. ACL*, vol. 1, 2015, pp. 795–804.
- [34] C. Li, Y. Duan, H. Wang, Z. Zhang, A. Sun, and Z. Ma, "Enhancing topic modeling for short texts with auxiliary word embeddings," *ACM Trans. Inf. Syst.*, vol. 36, no. 2, pp. 1–30, Sep. 2017.
- [35] *Raspberry Pi 4 Model B*. Accessed: Dec. 16, 2020. [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>
- [36] C. Cabrera, F. Li, V. Nallur, A. Palade, M. A. Razzaque, G. White, and S. Clarke, "Implementing heterogeneous, autonomous, and resilient services in IoT: An experience report," in *Proc. IEEE 18th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2017, pp. 1–6.
- [37] *Internet of Things APIS*. Accessed: Mar. 5, 2021. [Online]. Available: <https://www.programmableweb.com/category/internet-things/api>
- [38] *Owls-SLR: An Owl-S Service Profile Matchmaker*. Accessed: Jan. 2, 2021. [Online]. Available: <http://lpls.csd.auth.gr/systems/OWLS-SLR/datasets.html>
- [39] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Jan. 2010.
- [40] *MongoDB*. Accessed: Jan. 2, 2021. [Online]. Available: <https://www.mongodb.com/>
- [41] X. Wang, A. McCallum, and X. Wei, "Topical N-grams: Phrase and topic discovery, with an application to information retrieval," in *Proc. 7th IEEE Int. Conf. Data Mining (ICDM)*, Oct. 2007, pp. 697–702.
- [42] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical word embeddings," in *Proc. 29th AAAI Conf. Artif. Intell. (AAAI)*, 2015, pp. 2418–2424. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2886521.2886657>
- [43] W. Wang, P. M. Barnaghi, and A. Bargiela, "Probabilistic topic models for learning terminological ontologies," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 1028–1040, Jul. 2010.
- [44] A. Bakalov, A. McCallum, H. Wallach, and D. Mimno, "Topic models for taxonomies," in *Proc. 12th ACM/IEEE-CS Joint Conf. Digit. Libraries (JCDL)*, 2012, pp. 237–240.
- [45] D. Newman, A. Asuncion, P. Smyth, and M. Welling, "Distributed algorithms for topic models," *J. Mach. Learn. Res.*, vol. 10, pp. 1801–1828, Aug. 2009.
- [46] H.-F. Yu, C.-J. Hsieh, H. Yun, S. V. N. Vishwanathan, and I. S. Dhillon, "A scalable asynchronous distributed algorithm for topic modeling," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1340–1350.
- [47] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in Twitter to improve information filtering," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2010, pp. 841–842.
- [48] K. Li, J. Xie, X. Sun, Y. Ma, and H. Bai, "Multi-class text categorization based on LDA and SVM," *Procedia Eng.*, vol. 15, pp. 1963–1967, Jan. 2011.



**MOHAMMAD ABDUR RAZZAQUE** (Senior Member, IEEE) received the Ph.D. degree from UCD, Ireland. He is currently a Senior Lecturer with the School of Computing, Engineering and Digital Technologies, Teesside University. Before joining Teesside University, he worked as a Senior Research Fellow with Trinity College, Dublin, from October 2014 to January 2018 and a Senior Lecturer with UTM, Malaysia, from September 2011 to September 2014. His research interests include end-to-end IoT solutions and cybersecurity. He is also an Editor of the *International Journal of Distributed Sensor Networks* and *IoT Journal*.

• • •