Noname manuscript No.
(will be inserted by the editor)

An Empirical Study of IoT Topics in IoT Developer Discussions on Stack Overflow

Gias Uddin, Fatima Sabir, Yann-Gaël Guéhéneuc, Omar Alam, Foutse Khomh

Abstract Internet of Things (IoT) is defined as the connection between places and physical objects (i.e., things) over the Internet via smart computing devices. It is a rapidly emerging paradigm that encompasses almost every aspect of our modern life, such as smart home, cars, and so on. With interest in IoT growing, we observe that the IoT discussions are becoming prevalent in online developer forums, such as Stack Overflow (SO). An understanding of such discussions can offer insights into the prevalence, popularity, and difficulty of various IoT topics. For this paper, we download a large number of SO posts that contain discussions about various IoT technologies. We apply topic modeling on the textual contents of the posts. We label the topics and categorize the topics into hierarchies. We analyze the popularity and difficulty of the topics. Our study offers several findings. First, IoT developers discuss a range of topics in SO related to Hardware, Software, Network, and Tutorials. Second, secure messaging using IoT devices from the Network category is the most prevalent topic, followed by scheduling of IoT script in the Software category. Third, all the topic categories are evolving rapidly in SO, i.e., new questions are being added more and more in SO about IoT tools and techniques. Fourth, the "How" type of questions are asked more across the three topic categories (Software, Network, and Hardware), although a large number of questions are also of the "What" type: IoT developers are using SO not only to discuss how to address a problem related to IoT, but also to learn what the different IoT techniques and tools offer. Fifth, topics related to data parsing and micro-controller configuration are the most popular. Sixth, topics related to multimedia streaming and Bluetooth are the most difficult. Our study findings have implications for all four different IoT stakeholders: tool builders, developers,

Gias Uddin (Corresponding Author)
University of Calgary, E-mail: gias.uddin@ucalgary.ca
Fatima Sabir
Concordia University, E-mail: fatima.sabir@concordia.ca
Yann-Gaël Guéhéneuc
Concordia University, E-mail: yann-gael.gueheneuc@concordia.ca
Omar Alam
Trent University, E-mail: omaralam@trentu.ca
Foutse Khomh
Polytechnique Montréal, E-mail: foutse.khomh@polymtl.ca

educators, and researchers. For example, IoT developers and newcomers can use our findings on topic popularity to learn about popular IoT techniques. Educators and researchers can make more tutorials or develop new techniques to make difficult IoT topics easier. IoT tool builders can look at our identified topics and categories to learn about IoT developers' preferences, which then can help them develop new tools or enhance their current offerings.

Keywords Stack Overflow, IoT, Topic Modeling

1 Introduction

The Internet of Things (IoT) is a rapidly emerging paradigm that is defined as the connection between places and physical objects (i.e., things) over the Internet via smart computing devices [11, 38]. This paradigm is now pervasive in almost every aspect of our life, such as smart homes, cars, voice-enabled home assistants, and so on [4, 76]. According to Texas Instruments [27], the number of "smart" connected devices was 5 billions in 2013, to become 50 billions by 2020 (i.e., 1,000% increase in seven years). Thus, it is not surprising that interest in IoT technologies is growing among developers to develop IoT systems using IoT architectures, techniques, and tools [56, 111].

With interest in the IoT growing, we observe that discussions related to this paradigm are becoming increasingly prevalent in online developer forums, such as Stack Overflow (SO). An understanding of these discussions can offer insights into the prevalence, popularity, and difficulty of various IoT topics. SO is a large online community where millions of developers ask and answer questions. To date, there are around 120 million posts and 12 million registered users on SO [67].

Several research works have been conducted on SO posts, e.g., discussions on big data [12], concurrency [3], programming issues [15], blockchain development [108], microservices [14], or security [118]. However, to the best of our knowledge, there is no research analyzing discussions related to the IoT, although such insight can complement existing IoT literature—which has mainly used surveys to understand IoT developers' issues and needs [4, 11, 38].

Consequently, in this paper, we analyze over 53,000 IoT-related posts on SO and apply topic modeling [21] to understand the discussion topics related to the IoT. Thus, we answer the following four research questions:

RQ1. What IoT topics are discussed by developers on SO? The rapid advances of the IoT paradigm has led to the creation of many architectures, techniques, and tools to support IoT-based software development. It is thus important to understand how IoT developers are using these and what challenges they face. Previous research works on the IoT attempted to learn about the IoT using surveys; they did not consider IoT developers' discussions and real-world experience.

We apply topic modeling on our SO IoT dataset to identify topics in the developers' discussions about the IoT. We find a total of 40 topics in the 53,000 IoT-related posts. We group the 40 topics into four high-level categories: Software, Network, Hardware, and Tutorials. Software has the highest coverage of questions and topics (41.3% of questions, 19 topics), followed by Network (33.3% of questions, 11 topics), Hardware (20% of questions, 8 topics), and Tutorials (5.3% of questions, 2 topics). Out of the 40 topics, discussions related to Secure Messaging

among IoT devices and other devices are found in the greatest number of questions (5.8%), followed by Script Scheduling (4.8%) about the creation, scheduling, and deployment of batch jobs on IoT devices.

RQ2. How do the IoT topics evolve over time? Our findings from RQ1 show that IoT topics are quite diverse. However, the IoT is an emerging paradigm with new architectures, techniques, and tools regularly proposed, used, and retired across the four high-level categories (Software, Network, Hardware, and Tutorials). Developers' interests in the categories and their topics are likely to vary over time. Hence, we want to determine how IoT topics change in developers' discussions.

We compute the absolute impact of each category by determining how many new questions are added every month. We find that the absolute impact of each category is increasing in SO and this trend accelerated since 2014, possibly because many IoT software and hardware were introduced around then, e.g., the hugely popular Raspberry Pi A+ and B+ models. We report that changes in the IoT categories are correlated with the availability of new IoT software and hardware.

We also compute the relative impact of the categories, i.e., how many new questions are added in a month to a category relative to the other categories. We find that more questions related to Software are asked each month in comparison to questions in other categories. This trend is more nuanced starting from 2016 with more discussions occurring across all categories, possibly because of the entrance of Mozilla and other such renown companies into the IoT market.

RQ3. What types of questions do IoT developers ask about IoT topics? Given that the discussions in SO about the IoT are technical by definition, we can learn about the IoT developers' challenges by analyzing their discussions in SO. To learn about these challenges, we must distinguish the types of questions that developers are asking across the IoT topics. We manually label a large, statistically-significant sample of 1,439 questions in our dataset. We distinguish question types using the four categories previously used by Abdellatif et al. [1] for SO questions: How, What, Why, and Other.

We find that more than 47% of the questions are of type How, i.e., IoT developers ask and discuss how to complete their development tasks using IoT architectures, techniques, or tools. Also, about 38% of the questions are of type What, i.e, about *what* architectures, techniques, or tools to use. Around 20% of the questions are about the Why, i.e., about the specific behaviour of some IoT architectures, techniques, or tools. Questions of type How are most prevalent in the Hardware category (58% of all questions), followed by Network (51.5%), and Software (41.6%). These findings suggest that IoT developers question how to complete tasks, especially when these tasks involve IoT hardware or networking.

RQ4. How do the popularity and difficulty of IoT topics vary? Our findings from RQ1 show that there are many IoT topics discussed in SO while those from RQ3 show that many of the questions are of types How and What for the Software, Network, and Hardware categories. We conclude that what to do and how to do it is challenging for IoT developers. Yet, all topics cannot be equally popular or difficult. Popularity and difficulty help prioritizing research and industry efforts into particular topics: newcomers can focus on popular topics while experts can devise new architectures, techniques, or tools to ease certain topics.

Thus, for each IoT topic, we analyze four popularity metrics (score, view and favorite counts, # of answers received) and three difficulty metrics (% of questions

with accepted answer, average hours before an accepted answer, and % of questions without answer). We report that the topic Data Parsing in the Software category is most popular while the topic Multimedia Streaming in the same category is the most difficult. The Bluetooth Low Energy (BLE) topic in the Network category is the second most difficult, yet it is only the eighth most popular topic. Seven out of the nine difficulty and popularity metrics are negatively correlated: more difficult topics are generally less popular topics.

Conclusion. Our findings have implications for the four IoT stakeholders: builders, developers, educators, and researchers. Builders could look at the topics and categories to learn and support IoT developers' preferences, with new or enhanced offerings. Developers could use our findings on topic popularity to learn popular IoT techniques. Educators and researchers could make more tutorials or propose new architectures, techniques, or tools to ease difficult IoT topics.

Replication Package. All our data is available at https://github.com/disa-lab/IoTTopic.

Paper Organization. Section 2 discusses background and related work. Section 3 describes the details of our data collection and topic modeling processes. Section 4 answers our four research questions. Section 5 compares our topic modeling results with those of similar studies done on SO for other subjects, e.g., big data, and discusses their implications. Section 7 discusses threats to the validity of our results. Section 8 concludes the paper.

2 Background and Related Work

The IoT is gaining popularity because of the continuous involvement of a wide community. There are seemingly unlimited applications of the IoT in every sphere of life while developing IoT systems comes with many challenges. Thus, developers have been discussing IoT-related topics on various forums, in particular the well-known Stack Overflow Question and Answer Web site. An understanding of these discussions requires modeling their topics. We now discuss relevant studies associated to topic modeling and their applications in software-engineering research.

2.1 Studies Related to the IoT

With the IoT paradigm rapidly evolving, literature on the IoT has so far used surveys to understand this paradigm. Surveys exist on IoT architectures, techniques, and tools [4, 11, 87, 120], underlying middleware solutions (e.g., Hub) [26], big data analytics for smart devices [56], the design of secure protocols [4, 36, 46, 122], their applications on diverse domains (e.g., eHealth [61]), industrial adoption of IoT [50, 110], and the evolution and future of the IoT [27, 76, 89].

These surveys reported on various trends in the IoT, in both academia and industry, including connectivity among different devices [25, 57, 62, 106], cloud computing, cybersecurity, and big data [25], life cycle model for IoT systems [77], problems faced by users during the initial stage of usage [77, 106]. Claims were also made on the benefits of the IoT for different industries, from health care to manufacturing [90], from agriculture [60] to robotics [43].

However, none of these previous works specifically focused on issues faced by developers while developing IoT systems, i.e., working with concrete computing tasks, like *temperature sensors on Arduino* [22].

2.2 Topic Modeling

Topic modeling encompasses methods, techniques, and tools to manage, understand, and summarize large collections of textual information [54]. It helps reveal hidden dependencies among different patterns associated to specific topics in sets of textual documents [105]. It also provides topics that are groups of words best representative of the information embedded in the documents [97].

Topic modeling originated from generative probabilistic modeling in the 1980's [54]. It proposes that the interaction between observed and unobserved variables and probabilistic relationship between observations help generating meaningful and representative topics for a given dataset [94]. The first technique for topic modeling was the *TF–IDF* reduction scheme, normally used for information retrieval [85]. An alternative approach to *TF–IDF* was a dimensionality-reduction method developed by Deerwester et al. [32], named Latent Semantic Indexing or Latent Semantic Analysis (LSI/LSA). LSI uses the *TF–IDF* matrix factorization with the help of Singular Value Decomposition (SVD).

Topic modeling is particularly useful for textual document. However, it has also been applied to environmental data [37], bioinformatics data [54], and social-science data [40]. Some of its applications include structuring databases of different journals and articles into unique groups with similar focus [19], grouping social-media users by similar post content [40], and categorizing genomic data based on similar sequence structure [53].

In this paper, we apply topic modeling to SO posts related to the IoT. Our purpose is to create abstractions of the discussions in the forms of sets of topics. We use Latent Dirichlet Allocation (LDA) [21] to obtain topics. LDA is a probabilistic topic model. It is widely used in software-engineering research for modeling topics in software repositories [29, 96, 97]. The topics produced by LDA are less likely to overfit the documents and are more interpretable than those obtained from other algorithms, such as Probabilistic Latent Semantic Analysis (PLSA) [20, 21, 94].

In topic modeling, a *topic* is a collection of *frequently* co-occurring *words* in a set of textual *documents*. In our analysis, each document is a post from SO. Let $C = \{p_1, p_2, \ldots, p_n\}$ be our initial corpus of n SO posts. LDA takes as input C and the number m of topics that we want to obtain from the corpus. Each topic is defined by a probability distribution over all the unique words in C. LDA uses two Dirichlet priors, α and β, to generate a topic distribution. The prior α is used to produce the topic post distribution θ_i for each post p_i. The prior β produces the topic word distribution ϕ_j for each word in C. θ_{tp} is the probability of topic t for post p and ϕ_{tw} that of word w in topic t. The probability of generating word w in post p can be computed as follows: $p(w|p) = \sum_{t=1}^{T} \theta_{tp}\phi_{tw}$. The output of LDA is a set of m topics $T = \{t_1, t_2, \ldots, t_m\}$ with two matrices: 1. W: A topic word matrix, i.e., the most probable words for each topic. 2. D: A topic post matrix, i.e., the most probable topics in each post. We refer the readers to the original paper by Blei et al. [21] for details about LDA.

2.3 Topic Modeling in Software Engineering

Very large volume of data is nowadays available thanks to existing software repositories. Yet, it is estimated that 80% to 85% of the data in these repositories are unstructured [23], e.g., textual documentation, bug reports, or use cases [59]. The presence of unstructured data, although an opportunity for research works on program comprehension, traceability-link recovery, and feature location [59], makes it difficult, but important, to identify topics in this data.

Topic modeling has been increasingly used to mine unstructured software data [59]. It has been applied in the context of various software-engineering activities, like program comprehension, and for different domains, from object-oriented programming to the IoT. For example, Program Feature Network (PFN) were used to identify semantic features in object-oriented systems at class level [55] while topic XP provided insights about software systems by extracting information from source-code identifiers using LDA [86].

Feature identification, or concept location, can identify and create links between documentation describing requirements and the source code that implement these requirements [33]. Poshyvanyk et al. [74] combined formal concept analysis with LSI to map concepts in textual change requests with relevant parts of a source code. Revelle et al. [34] performed feature location using dynamic analysis with the help of data fusion, LSI, and Web mining algorithms. Nie et al. [65] used LDA to locate interesting parts of source code with a measure of topic cohesion based a software dependency network.

Topic modeling was also used to identify concerns in source code [64]. Andrzejewski et al. [7] proposed an approach based on Delta Latent Dirichlet Allocation (DLDA) to identify two types of topics in program execution traces: normal usage and buggy execution. Statistical topic modeling technique was also used to identify concerns in software systems [28]. Hu et al. [41] used information from relation strength to predict defect prone source code.

Xie et al. [114] proposed Dretom for the recommendation of a developers to solve a bug. Dretom uses a topic model and the developers' development experience in resolving bugs. Statistical author-topic models were also used as a recommendation system for developers [51]. Zhang et al. [122] combined topic modeling with developer role as bug reporter and–or assignee to identify developers' major knowledge areas. Yang et al. [117] suggested an approach for the recommendation of bug fixes using the similarity among bug-report topics. Others also used relational topic modeling to recommend the Move Method refactoring [16, 17].

Topic models were recently used to understand software logging [49]. Other applications include concept and feature location [30, 75], traceability link recovery [10, 78], source-code history [41, 99, 100], code search [101], refactoring [17], to explain software defect [28], and to ease various maintenance tasks [95, 96].

Our motivation to use topic modeling to understand IoT discussions stems from this previous work showing that topic modeling is useful in software-engineering research and that textual documents can be approximated by their topics [29, 96, 97]. We follow recommended parameterization of topic modeling for software engineering tasks [68, 69].

SO posts were the subjects of several recent papers to study various aspects of software development using topic modeling, such as what developers are discussing in general [15], or about a particular problem, e.g., concurrency [3], big

data [12], chat-bot development [1]. Topic modeling was also used to provide valuable contributions to cloud services [63] and Web services [24, 109]. Researchers used deep learning techniques for modeling topics in big data [70, 123] and block chain [88]. SO was also analysed for trends in reference architectures of block chain [108]. We also noticed the use of industry-related forums, especially about block chain [44, 52].

Recently, Aly et al. [5] examined questions related to IoT and Industry 4.0 on SO. Similarly to the work presented here, they applied topic modeling to identify the topics discussed in the studied questions. Although they assessed the popularity and difficulty of the topics, they did not examine the evolution of these topics over time. They also did not investigate the types of questions asked by developers working on IoT systems. Moreover, their study mainly focused on understanding the industrial challenges of the IoT technology, while our work wants to understand the specific challenges faced by developers implementing concrete IoT systems. Our study also conducts an in-depth analysis of IoT related questions to understand how the popularity and difficulty of IoT topics are correlated.

3 Data Collection and Topic Modeling

We now explain our data collection process to obtain IoT-related posts in SO in Section 3.1. We then discuss how we pre-processed and applied topic modeling on this data to find IoT topics in Section 3.2.

3.1 Data Collection

We follow three steps to collect IoT-related SO posts: 1. Obtain a SO dataset, 2. Identify IoT tagset in the dataset, 3. Identify IoT-related posts in the dataset based on the IoT tagset. We describe these steps in the following.

Step 1. Obtain a SO Dataset. We chose Stack Overflow (SO) for our study because SO is one of the most popular online Q&A Web sites for developers to discuss diverse topics related to software and hardware development [71, 102]. We downloaded the SO data dump [91] of September 2019, which was the latest dataset available during our analysis. The dataset includes all posts for 11 years between 2008 and September 2019 for a total of 46,767,375 questions and answers. Out of those Q&A, around 40% are questions and 60% are answers. Around 34% of the answers are accepted. A total of 11,337,789 users participated in the discussions.

Each post in the dataset contains the following information: 1. Content, including textual content and code example, 2. Creation and edition times, 3. Score, favorite, and view counts, 4. User ID who created the post 5. User-provided tags given to the post. An answer to a question is flagged as accepted if the user who asked the question chose this answer. A question can have between 1 and 5 tags.

Step 2. Identify IoT Tagset in the Dataset Not all the posts in the dataset relate to the IoT. We thus must determine the posts that contain IoT-related discussions. We use the user-defined tags assigned to the questions. We determine the set of tags that could mark a discussion as related to the IoT. We adapt the following approach from Yang et al. [119]: 1. We identify three initial, popular IoT-related tags in SO. We denote those as $\mathcal{T}_{init}$. 2. We collect posts tagged with

$\mathcal{T}_{init}$ and analyze the other tags given to these posts. The final set $\mathcal{T}_{final}$ contains 75 tags. We discuss each step below.

1. *Identify Initial IoT Tags.* Intuitively, a significant number of IoT-related posts in SO should be labeled with "iot". On September 2019, we thus searched for "iot"-tagged posts. SO search engine returned posts tagged with "iot" and a set of 20 other tags relevant to these posts, such as "raspberry-pi", "arduino", "windows-10-iot-core", "python", etc. These are tags that most frequently co-occur with the "iot" tag. These 20 tags can be broadly categorized as: (a) Tags with "iot" suffix/prefix, e.g., "windows-iot", (b) Tags related to the usage of two predominantly-used IoT technologies, "arduino" and "raspberry-pi", and their various incarnations across domains, e.g., "homekit". We thus consider the following tags in our initial set $\mathcal{T}_{init}$: (a) "iot" or any tag with "iot", e.g., 'azure-iot-hub', (b) "arduino", (c) "raspberry-pi". These tags are sensible because Arduino and Raspberry PI are the two most popular devices to develop IoT-related applications. Both families of devices underwent rapid evolution through multiple versions, such as the Raspberry PI 2.
   As we noted above, the three initial tags are selected by using the Stack Overflow (SO) search engine. We started the search by looking for questions in SO that are tagged as 'iot'. The details of the tag search engine are not shared by Stack Overflow. However, we find discussions in Stack Exchange meta sites where users queried about the specifics of suggesting related tags. For example, the developer Prashant asked this question *"How does Stack Overflow suggest related tags?"*[1], while the developer user1306322 asked another similar question as *"What are these tags related to on the Newest Questions page?"*[2]. According to the answers posted to these questions, related tags are those that frequently appeared together with one tag. SO also has an API endpoint[3] to search for related tags, where it describes the functionality as *"Returns the tags that are most related to those in tags. Including multiple tags in tags is equivalent to asking for "tags related to tag #1 and tag #2" not "tags related to tag #1 or tag #2". Count on tag objects returned is the number of questions with that tag that also share all those in tags."* When we searched in early 2020, the SO search engine returned 20 other tags related to 'iot' tag (the SO engine returned 25 relevant tags in early 2021). As we explained in Section 3.1, not all the 20 tags are specific to IoT. For example, one relevant tag was 'python' which contains general python programming questions. Therefore, it is not practical to manually analyze each question tagged as 'python' to isolate IoT-related questions, unless the question is also not tagged as 'iot'. On the other hand, it is possible that some IoT developers used more specific tags other than 'iot' to ask IoT-related questions. That means, we cannot simply rely on questions tagged as "iot" in SO to find all IoT-related discussions. Therefore, we manually analyzed each of the 20 related tags and decided to use three initial tags: iot, arduino, and raspberry-pi. We then applied our following tag-expansion algorithm on the entire SO dump by using the three initial tags as seeds. The tag-expansion algorithm is previously used to collect other domain-specific posts, e.g., to find big-data [13], concurrency [3], mobile apps [83], chat-bot posts [1], etc.

[1] https://meta.stackexchange.com/questions/235092
[2] https://meta.stackexchange.com/questions/186910
[3] https://api.stackexchange.com/docs/related-tags

2. *Determine Final IoT Tagset.* Intuitively, besides the initial tags in $\mathcal{T}_{init}$, there can be many IoT-related tags in SO posts that developers frequently use to tag IoT-related questions. Let the entire SO dataset be denoted by $\mathcal{D}$. First, we use the tags in $\mathcal{T}_{init}$ to find IoT-related posts. Second, we extract all questions $\mathcal{P}$ in $\mathcal{D}$ labeled with at least one of the tags in $\mathcal{T}_{init}$. Third, we extract all tags $\mathcal{T}_{\mathcal{A}}$ in $\mathcal{P}$. Not all the tags in $\mathcal{T}_{\mathcal{A}}$ may correspond to IoT topics (e.g., "python"). Therefore, following previous work [13,119], we filter irrelevant tags and finalize IoT-related tags in $\mathcal{T}$ from $\mathcal{T}_{\mathcal{A}}$ as follows.
For each tag t in $\mathcal{T}_{\mathcal{A}}$, we compute its significance and relevance for $\mathcal{P}$:

$$\text{Significance } \mu(t) = \frac{\#\text{of Questions with tag } t \text{ in } \mathcal{P}}{\#\text{of Questions with tag } t \text{ in } \mathcal{D}} \tag{1}$$

$$\text{Relevance } \nu(t) = \frac{\#\text{of Questions with tag } t \text{ in } \mathcal{P}}{\#\text{of Questions in } \mathcal{P}} \tag{2}$$

A tag t is significantly-relevant to the IoT if $\mu(t)$ and $\nu(t)$ are higher than some specific thresholds. Our 49 experiments with a broad range of threshold values of $\mu = \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35\}$ and $\nu = \{0.001, 0.005, 0.01, 0.015, 0.02, 0.025, 0.03\}$ show that $\mu = 0.3$ and $\nu = 0.001$ yield the most number of relevant IoT tags, while reducing false positives. These threshold values are consistent with previous work [3, 13, 119].

We use the following steps to find our list of IoT tags using the above two variables.

1. We collect all the tags that co-occurred in SO with our three initial IoT tags. This resulted in a total 5,672 tags.
2. We then collect a subset of the 5,672 tags based on a threshold value pair. For example, for $\mu = 0.3$ and $\nu = 0.001$, we obtain 60 tags. We denote the tags as 'Tags Recommended'by our two threshold value pairs. For each tag in the list 'Tags Recommended', we manually analyze the tag to determine if it is actually related to IoT. We do this determination by consulting the description of the tag in SO. For example, we do not consider this tag as relevant to IoT: 'iota'. This tag is described in SO tag wiki [93] as *"The iota function is used by several programming languages or their libraries to initialize a sequence with uniformly increasing values."* However, we consider the following tag as relevant to IoT: 'omxplayer', which is described in SO tag wiki as *"Omxplayer is a video player specifically made for the Raspberry Pi's GPU."* The manual analysis yields 53 IoT tags.
3. We repeat step 2 above for each of the 49 pairs from $\mu = \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35\}$ and $\nu = \{0.001, 0.005, 0.01, 0.015, 0.02, 0.025, 0.03\}$. In Figure 1, we show the relevance of tags along the 49 experiments. We show one stacked bar per experiment (i.e., one pair of μ and ν). For each bar, we show the total number of tags returned (i.e., Total Recommended) by the threshold value pair and the total number of tags that we considered as IoT tags based on manual analysis (i..e, Total Relevant). For the value pair $\mu = 0.3$ and $\nu = 0.001$, we find 53 tags as relevant out of the 60 recommended tags(88.3%). If we lower the threshold values, we do not have an increase in the number of relevant tags, but some new tags appeared that were not found based on the value pair $\mu = 0.3$ and $\nu = 0.001$. With lower threshold value pairs, the number of false positives increases. With higher threshold value pairs, we lose many relevant IoT tags.

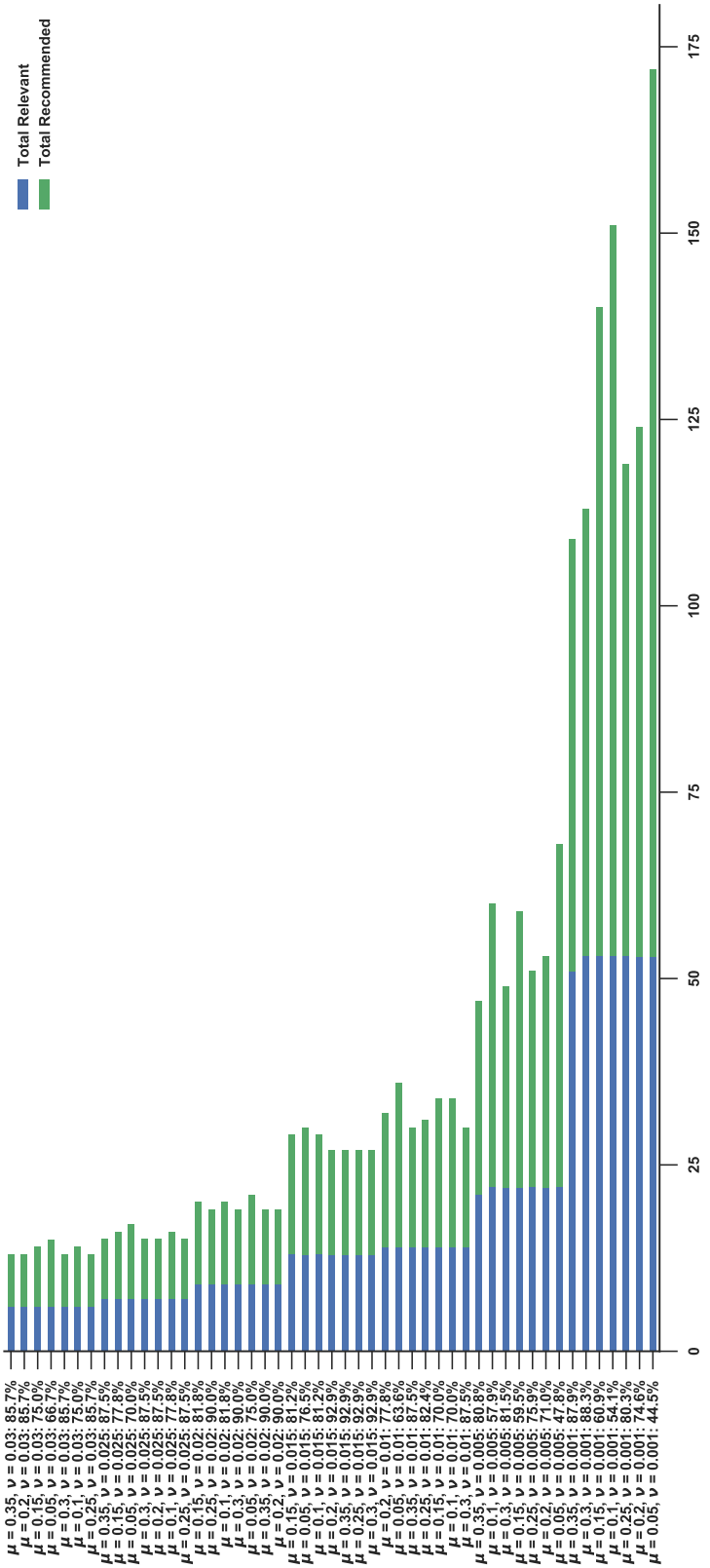

Fig. 1: Total recommended vs relevant IoT tags based on different μ and ν values in SO based on three initial IoT tags

4. We compile our final list of IoT tags by collecting all the tags found as relevant from our 49 experiments. The final set of tags contain total 75 IoT tags that are found as relevant through our manual analysis in the 49 experiments. These 75 tags cover a wide range of technologies and tools supporting the IoT in software development. The tools from major software vendors are offered by their IoT-platforms, e.g., "aws-iot" from Amazon, "google-cloud-iot" from Google, "watson-iot" from IBM, "azure-iot" and "windows-iot" from Microsoft, and so on. A variety of IoT-based network technologies are also present in the tags, such as "lora" or "xbee". The IoT is supported by several notable platforms and SDKs, as evidenced by the tags "johny-five", "raspian", "attiny", etc. The tags are listed below. The list of all 5,672 tags with their μ and ν values are provided in our online replication package.
   $\{T_{arduino}, T_{iot}, T_{raspberry-pi}\}$ = { arduino, arduino-c++, arduino-due, arduino-esp8266, arduino-every, arduino-ide, arduino-mkr1000, arduino-ultra-sonic, arduino-uno, arduino-uno-wifi, arduino-yun, platformio, audiotoolbox, audiotrack, aws-iot, aws-iot-analytics, azure-iot-central, azure-iot-edge, azure-iot-hub, azure-iot-hub-device-management, azure-iot-sdk, azure-iot-suite, bosch-iot-suite, eclipse-iot, google-cloud-iot, hypriot, iot-context-mapping, iot-devkit, iot-driver-behavior, iot-for-automotive, iot-workbench, iotivity, microsoft-iot-central, nb-iot, rhiot, riot, riot-games-api, riot.js, riotjs, watson-iot, windows-10-iot-core, windows-10-iot-enterprise, windows-iot-core-10, windowsiot, wso2iot, xamarin.iot, adafruit, android-things, attiny, avrdude, esp32, esp8266, firmata, gpio, hm-10, home-automation, intel-galileo, johnny-five, lora, motordriver, mpu6050, nodemc, omxplayer, raspberry-pi, raspberry-pi-zero, raspberry-pi2, raspberry-pi3, raspberry-pi4, raspbian, serial-communication, servo, sim900, teensy, wiringpi, xbee}

Step 3. Identify IoT-related Posts in the Dataset based on the IoT Tagset. Our final dataset consists of all posts tagged with at least one of the 75 tags. We consider that a SO question is an IoT question if it is tagged by one or more of the tags from $\mathcal{T}$. Based on the 75 tags, we found a total of 81,651 posts (questions and answers) in our SO dataset, out of which around 48% are questions (i.e., 39,305) and 52% (42,346) are answers. Around 33% of the answer (13,868) are accepted. Following previous work [13, 15, 83], we only consider questions and accepted answers. We exclude unaccepted answers to avoid noise and reduce the size of the dataset. Thus, the final dataset $\mathcal{B}$ consists of a total of 53,173 posts (39,305 questions, 13,868 accepted answers).

3.2 Topic Modeling

We follow three steps to produce IoT topics from IoT posts in the dataset $\mathcal{B}$: 1. Pre-process IoT post text, 2. Find an optimal number of topics, 3. Generate the IoT topics. We discuss the steps below.

Step 1. Pre-process IoT Post Text. We pre-process the posts in our dataset $\mathcal{B}$ to reduce noise. Specifically, we follow the noise reduction steps adopted in previous work [13, 15, 119]. First, we remove all the paragraphs/contents in a post that are non-text blocks, e.g., code snippets marked with $< code >< /code >$

or other HTML tags used to mark non-text blocks, such as $< p >< /p >$ and $< a >< /a >$. Second, we remove stop words (e.g., "a", "an", "the", etc.), numbers, punctuation marks, non-alphabetical characters, etc. We use the set of all stop words from NLTK [66] and MALLET [6]. This is a usual step in the processing of natural-language texts to ensure that the modeling focuses on the most informative content. Third, we apply Porter stemming [73] to obtain the roots of the words, which can increase the contextual understanding of some content by enhancing similarity while preserving the diversity in the content. For example, "configuration", "configured", "configure" are all reduced to "configur".

Step 2. Find an Optimal Number of Topics. To generate topics, we use the Latent Dirichlet Allocation (LDA) algorithm [21] available in the MALLET library [6]. Given as input the pre-processed posts in our dataset $\mathcal{B}$, the algorithm outputs a list of topics to group the posts into K topics. We use the standard practice to pick the optimal number K of topics as proposed by Arun et al. [8].

This standard practice suggests that the optimal number of topics will increase the measurement of *coherency* among the topics, i.e., the more coherent the topics, the better the topics can encapsulate the underlying concepts. Following previous work [103], we use the standard c_v metric originally proposed by Röder et al. [81] to determine the coherence, which is available in the Gensim package [79].

We run MALLET LDA on our dataset for varying number of K to obtain the topic model that has the best coherence score. Following previous work [3, 12, 15, 18, 83, 119], we use standard values of $\alpha = 50/K$ and $\beta = 0.01$ for the two hyper-parameters of LDA. First, we apply MALLET LDA on our dataset $\mathcal{B}$ for $K = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70\}$. Following [12], we run each model with 1,000 iterations. Second, for each K, we compute the coherence score of the produced topic model. A higher coherence score indicates better separation of the topics. Third, we pick the topic model that had the best coherence score. Our analysis shows that the coherence scores reach a maximum value of 0.7 for $K = 50$. We thus pick 50 as the optimal number of topics.

Step 3. Generate the IoT Topics. We generate 50 topics using the dataset and parameter above and the obtained topic model. For each topic, the model offers the following information: 1. **Words**. A list of the top N words explaining the topic and the probability of each word, which denotes the relative defining power of the word for the topic. We collect the top 30 words per topic. 2. **Posts.** A list of the M posts associated with the topic. Each associated post is given a correlation value between 0 and 1. The higher is the correlation of a post, the more "on topic" the post is. Following previous work [107], we assign the posts with the highest correlation values to the topic.

4 Empirical Study

In this section, we answer our four research questions:

1. What topics are discussed by IoT developers in (SO)? (Section 4.1)
2. How do the IoT topics evolve over time? (Section 4.2)
3. What types of questions are asked across the IoT topics?(Section 4.3)
4. How do the popularity and difficulty of the topics vary? (Section 4.4)

The first two research questions provide insights into the types and evolution of the IoT-related topics discussed in SO. The other two questions report on the IoT developers' challenges.

4.1 What IoT Topics Are Discussed by Developers in SO? (RQ1)

4.1.1 Motivation

The rapid progress of the IoT paradigm motivated the creation of many architectures, techniques, and tools to support IoT software development. It is thus important to understand how IoT developers are using these architectures, techniques, and tools and what challenges they face. As noted in Section 2, the literature on the IoT has so far used surveys, which did not include the developers' questions and answers and their real-world usage of the IoT. As such, an understanding of the topics of IoT developers' discussions in SO can be useful to learn the types of challenges they face.

4.1.2 Approach

To understand a topic, we must provide a label for it, which summarizes the underlying concepts of the topic. Following previous work on topic labeling [1, 3, 12, 83, 119], we use an open card sorting approach [42] to manually label each topic. In open card sorting, there is no predefined label for a topic, because such a label is identified during an open coding process. To label a topic, we use two types of information: 1. The list of top words in the topic, 2. A list of 20-30 randomly-selected questions associated with that topic.

The first and fourth author participated in the labeling process. They have Ph.D.s related to empirical software engineering and software design. During the card-sorting process, the coders assigned a label to each topic by discussing with each other. The coders iterated through the labeling of each topic, until an agreement was reached. In total, more than 20 iterations were required to reach an agreement, during which the coders discussed in person, by email, Skype, and phone.

After this labeling, we merged a number of topics, because they were similar but with different vocabularies, which LDA considered as different. For example, we merged Topics 22 and 25 into Library Installation Tutorial, because both topics contained discussions about software libraries and SDKs for IoT devices, but LDA put those in two topics due to the diverse range of libraries discussed in those topics. At the end, we obtained 40 distinct topics.

We revisited all topics to group those into higher categories. For example, the two topics Multimedia Streaming and Audio Processing are related to the processing of multimedia data through IoT devices. We thus put the two topics under the category Multimedia. We repeated this process multiple times to create increasingly higher categories, until no further higher categories were found. For example, these two topics related to Multimedia can be further grouped under Data Processing, i.e., the parsing and manipulation of various data types by IoT devices. This higher abstraction allowed other topics to be placed under Data Processing, e.g., Textual Data Parsing and Timezone Formatting. Similarly, we grouped

the different software-debugging topics under Software Troubleshooting. Finally, we further clustered the categories of topics into higher categories by revisiting each category. For example, both the categories 'Software Troubleshooting' and 'Data Processing' are clustered into a higher category 'Software', because the topics in those categories contain discussions about the usage and troubleshooting of software (e.g., API/SDK) in IoT devices. The entire process of finalizing the categories required multiple iterations. We developed a coding guide to ensure that the grouping was reproducible. We share the coding guide in our replication package.

4.1.3 Results

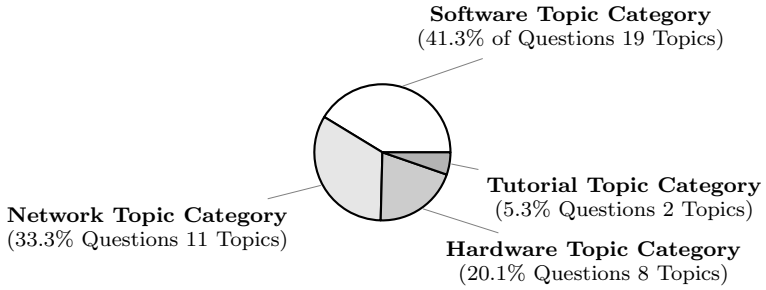

Fig. 2: Distribution of questions and topics per topic category

We found a total of 40 IoT topics. After labeling the topics, we grouped them into four high-level categories: **Software**, **Network**, **Hardware**, and **Tutorial**. Figure 2 shows the distribution of topics and questions in the four categories. Among the categories, Software has the highest coverage of questions and topics (41.3% of questions, 19 topics), followed by Network (33.3% of questions, 11 topics), Hardware (20% of questions, 8 topics), and Tutorial (5.3% of questions, 2 topics).

Figure 3 shows the 40 IoT topics by numbers of questions. On average, each topic is found in 983 questions. The topics are sorted, i.e., the topic with the greatest number of questions is placed first on the left. Out of the 40 topics, discussions related to Secure Messaging among IoT devices are found in the greatest number of questions (5.8%), followed by Script Scheduling (4.8%), i.e., the creation, scheduling, and deployment of batch jobs on IoT devices.

Figure 4 shows the 40 IoT topics grouped into the four categories, ordered based on the distributions of questions. For example, the topmost category Software is found in the most number of questions. Under each topic category, we group the topics into sub-categories. For example, the Software category has three sub-categories: 1. Platform Management, 2. Troubleshooting, 3. Data Processing. The topics under each sub-category are further divided into a number of groups. For example, there are seven topics under the sub-category Platform Management. The seven topics are grouped into three groups: 1. Service, 2. OS, 3. Virtualization. Each sub-category and each group are placed based on the distributions of their questions. For example, Platform Management is found the most (16.7% questions) under Software, followed by Troubleshooting (14.3% questions). Service is found the most (9.9%) under Platform Management.

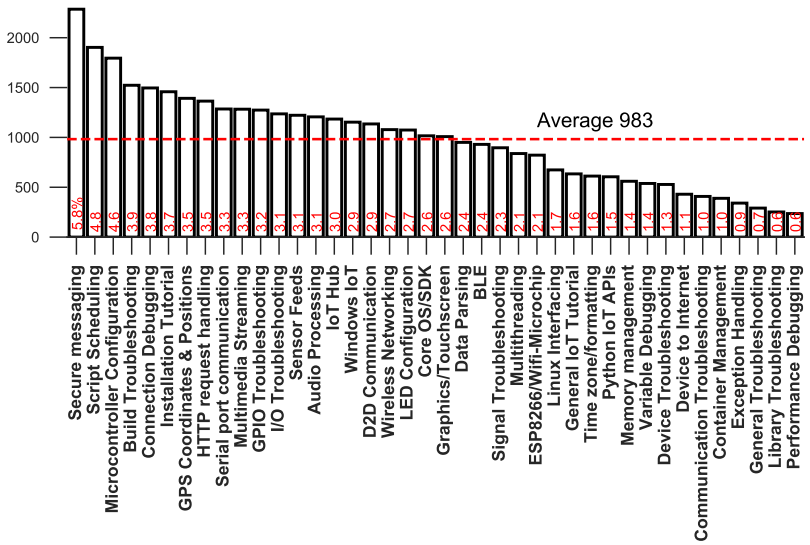

Fig. 3: Distribution of IoT Topics by Total #Questions

We now discuss the topics by the four categories below.

Software Related Topics. Out of the 40 topics, 19 topics belong to the Software category. These topics contain discussions about the usage, processing, and troubleshooting of IoT software, libraries, and data. The topics are grouped into three sub-categories: 1. Platform Management contains discussions related to IoT platforms (services, OS, SDKs, etc.), 2. Troubleshooting is about the debugging of source code and the underlying IoT platforms, 3. Data Parsing contains discussions about the processing of multimedia and textual contents through IoT devices.

Platform Management. Platform Management contains 16.7% of the questions and seven topics divided in three groups: 1. Service (9.9%) contains discussions related to the operations provided by IoT devices, 2. OS (4.3%) is related to the different operating systems for IoT devices, 3. Virtualization (2.5%) concerns the availability and usage of containers to operationalize IoT-based tasks.

The **Service** group has three topics: 1. Script Scheduling contains discussions related to scripts in different programming language (e.g., PHP, Python, Shell) that can do batch jobs, like using crontab on a Raspberry PI, e.g., $Q_{16488076}$[4]. 2. IoT Hub contains problems and solutions to cloud-based back-ends used by IoT devices, e.g., for processing Azure IoT hub messages, $Q_{49574377}$ or identity translation in Azure IoT edge gateways, $Q_{48786111}$. 3. Multi-threading discusses parallelization in IoT devices with diverse problems like processing of PWM signals within the limited CPUs of an IoT device, $Q_{9701895}$. The **OS** group contains two topics: 1. Core OS/SDK has discussions about different operating systems and SDKS, e.g., recently deprecated Android Things in $Q_{50932499}$, Eclipse Kura in $Q_{44944197}$, etc. 2. Linux Interfacing is about using Linux kernels, e.g., mount-

[4] Q_i and A_i denote a question or an answer in SO with ID i

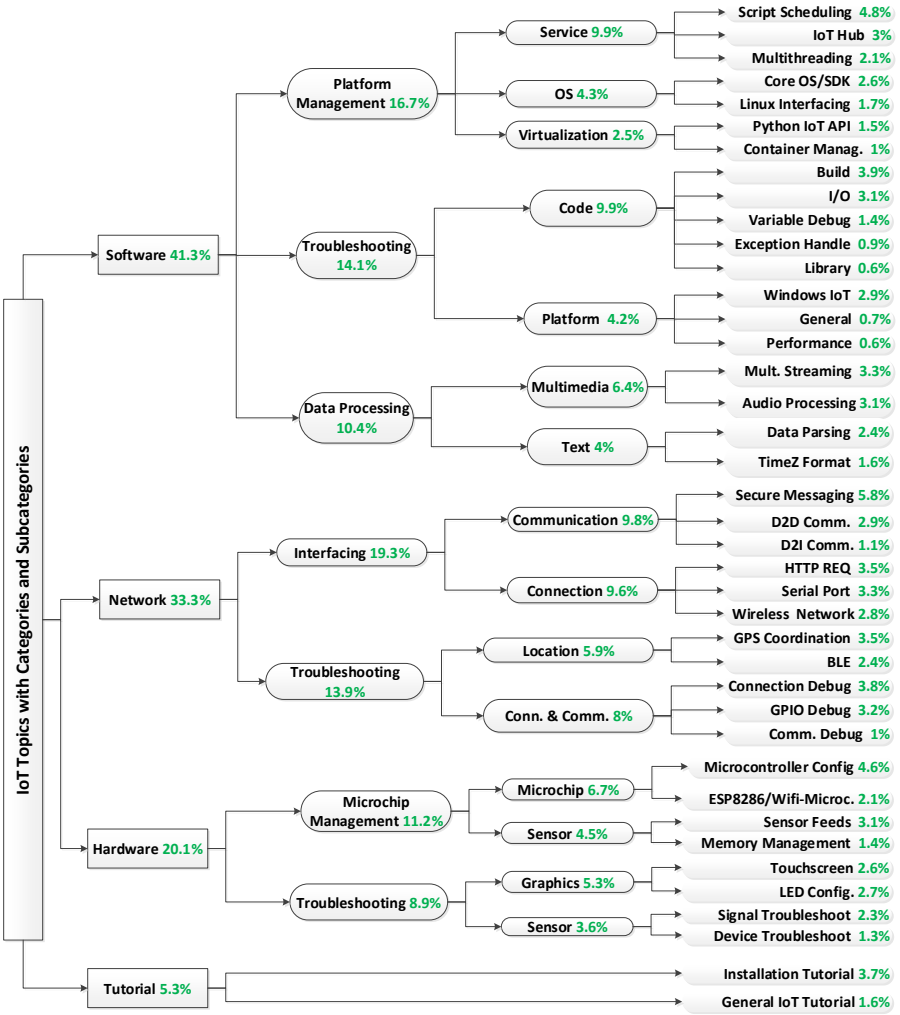

Fig. 4: IoT Topics with categories and subcategories

ing of USB drives in $Q_{42465326}$. The **Virtualization** group contains two topics: 1. Python IoT APIs, 2. Container Management, which discuss the availability of containers like Kubernetes and Traefik and cross-platform Python frameworks like Kivy, $Q_{47979205}$ and $Q_{41669449}$.

Software Troubleshooting. The Software Troubleshooting sub-category (14.1%) contains eight topics under two groups: 1. Code (9.9%) contains discussions about debugging source-code written for IoT devices, 2. Platform (4.2%) is related to performance troubleshooting or signals generated from IoT devices.

The **Code** group relates to the debugging of 1. Builds (3.9%) for IoT software, 2. I/O (3.1%) to process inputs and outputs, 3. Variable (1.4%) to encode long

strings, avoid overflow, $Q_{37479791}$, 4. Exception Handling (0.9%), like the crash of Android applications with Bluetooth, $Q_{38642352}$, 5. Library (0.6%) to troubleshoot the usagee of IoT libraries, e.g., loading LESetScanParameters and LESetScanEnable from Bluetooth devices, $Q_{36286698}$. In contrast, **Platform** is about troubleshooting the underlying platforms in three topics: 1. Windows IoT (2.9%), like the debugging of Universal Windows Platform (UWP) applications, e.g., $Q_{41176026}$, 2. General (0.7%), like the troubleshooting of Qt programs, $Q_{17994351}$), 3. Performance (0.6%), like segmentation faults in Raspberry PI, $Q_{38616768}$.

Data Parsing. The Data Parsing sub-category (10.4%) contains four topics in two groups: 1. Multimedia (6.9%) is about the processing of streaming contents, like Multimedia Streaming (3.3%) and Audio Processing (3.1%), 2. Text (4%) is related to the parsing of textual contents: Data Parsing (2.4%) and Timezone Formatting (1.6%).

Network Related Topics. A total of 11 topics are found under the Network category, which covers 33.3% of all questions in our dataset. The topics are clustered in two sub-categories: 1. Interfacing, i.e., communication and connection among IoT devices, 2. Troubleshooting of network properties and locations.

Interfacing. The Interfacing sub-category (19.3%) contains six topics under two groups: 1. Communication principles and techniques (9.8%), 2. Connection protocols and ports between IoT devices (9.6%).

The Communication group has three topics: 1. Secure Messaging (5.8%), 2. Device to Device (D2D) Communication (2.9%), 3. Device to the Internet (D2I) Communication (1.1%). The topic Secure Messaging contains the greatest number of questions among all IoT questions in our dataset. These questions cover principles, protocols, and issues related to authentication and authorization during messaging among and–or from IoT devices, e.g., set up advice and issues for AWS IoT connection using Cognito credentials, $Q_{55855320}$, $Q_{51184006}$). D2D Communication consists of communication between IoT client and online servers, e.g., between Ardunio client and PHP using cURL, $Q_{15621246}$), reading/writing to/from IoT devices, e.g., via serial port, $Q_{38627932}$, SD card $Q_{43703778}$, etc. D2I communication is about the setup of IoT devices as WiFi hot-spots, e.g., $Q_{51414572}$, or the controlling of an IoT device remotely, e.g., via PSP, $Q_{15001738}$, via Google IoT Core, $Q_{53888704}$, etc.

The Connection group contains three topics: 1. HTTP Requests (3.5%) to send/receive messages and controls, e.g., sending HTTP SOAP request to Sonos device with the NodeMCU firmware to connect IoT devices, $Q_{41897899}$, 2. Serial Port (3.3%) connection issues, e.g., reading between serial ports of IoT devices, $Q_{17566980}$, 3. Wireless Networking (2.8%) issues like creating a wireless mesh network with Raspberry PIs, $Q_{23437690}$, configuration of both static and dynamic IPs in the device, $Q_{31607892}$, etc.

Network Troubleshooting The Network Troubleshooting sub-category (13.9%) contains five topics in two groups: 1. Communication and Connection issues (8%) between IoT devices via GPIOs, 2. Location (5.9%) coordination.

The Communication and Connection issues contain three topics: 1. Connection Debugging (3.8%) of network connections, e.g., WiFi in Arduino, $Q_{38045838}$, failure to establish TCP/IP connection, $Q_{54548777}$, etc. 2. GPIO Debugging (3.2%) of the general purpose IO pins of the IoT devices, e.g., $Q_{15411746}$. 3. Communication

Debugging (1%) to test the communication between local/connected IoT devices and–or Cloud services, e.g., *" Google IoT Core Client for Android"* in $Q_{52948695}$.

The Location Troubleshooting group has two topics: 1. GPS Coordination (3.5%) is about the setup and configuration of location-based devices, e.g., accelerometer to move a robot, $Q_{39443604}$, 2. BLE (2.4%) is about the debugging of Bluetooth Low Energey, e.g., *" 9dof razor and BLE mini"*, $Q_{24788236}$.

Hardware Related Topics. We found eight hardware-related topics that covered 20.1% of all questions in SO IoT dataset. The topics are clustered around two sub-categories: 1. Microchip Management topics related to the configuration of micro-controllers and IoT sensors, 2. Troubleshooting of graphic cards and sensors.

Microchip Management The Microchip Management sub-category (11.2%) contains four topics under two groups: 1. Microchip Configuration (6.7%), 2. Sensor (4.5%) processing.

The Microchip Configuration group has two topics: 1. Microcontroller Configuration (4.6%), discussions about the connection between IoT devices and micro-controllers, e.g., between Arduino and Arduino Mega ADK in $Q_{21911256}$, 2. ESP8266/WiFi-Micro-controller (2.1%) contains discussions of the setup of IoT-based WiFi-microchips, e.g., ESP8266 Soft WDT reset, $Q_{48867927}$. The Sensor group has two topics: 1. Sensor Feeds (3.1%), 2. Memory Management (1.4%). The topics relate to processing sensor data and handling low-powered memory while analyzing sensor feeds, e.g., *" Import sensor data to RRDtool DB"* in $Q_{40309398}$.

Hardware Troubleshooting The Hardware Troubleshooting sub-category (8.9%) contains four debugging-related topics under two groups: 1. Graphics (5.3%), 2. Sensor (3.6%).

The Graphics group pertains to the debugging of: 1. Touchscreen (2.6%), 2. LED Configuration (2.7%). Many questions in Touchscreen are about the OpenGL library, such as the difficulty of getting value out of OpenGL ES 2 shaders on Raspberry PIs, $Q_{27754675}$. The questions related to LED Configuration are about the setup of LEDs and their controls via IoT devices, e.g., setup of Arduino Christmas lights in $Q_{40611990}$.

The Sensor group contains two topics: 1. Signal Troubleshooting (2.3%) is about debugging diverse IoT-based signals, e.g., V-USB button in $Q_{15870914}$, Arduino switch button in $Q_{25657310}$, etc., 2. Device Troubleshooting (2.1%) is related to the use of IoT devices, e.g., Raspberry PI 1B with the MJPG-Streamer, and a USB Web camera in $Q_{38663532}$.

Tutorials Related Topics. The Tutorials category covers 5.3% of the questions based on two topics: 1. Installation Tutorial (3.7%), 2. General IoT Tutorials (1.6%). The Installation Tutorials topic pertains to installing libraries, e.g., instructions to install the Mono framework on a Raspberry PI 3 running OpenHAB 2 in $Q_{32617411}$. General IoT Tutorials include the safe way to create data structures and strings from non-ASCII characters, $Q_{32071478}$. Overall, the Tutorials are not about questions or issues. Rather, the questions inquire about specific instructions and best practices asked mostly due missing information in the official tutorials/documentation.

Summary of RQ1. What IoT topics are discussed by developers in SO? We found 40 topics in our SO dataset of IoT discussions. The topics belong to four categories: Software, Network, Hardware, and Tutorials. The Software category has the greatest number of questions, followed by Network, Hardware, and Tutorial. Secure Messaging in the Network category is the topic with the greatest number of questions (5.8%), followed by Script Scheduling (4.8%) in the Software category. The discussions on troubleshooting IoT devices are prevalent across topics in the Software, Network, and Hardware categories.

4.2 How Do the IoT Topics Evolve Over Time? (RQ2)

4.2.1 Motivation

IoT topics and their categories have distinct features associated by the topic model. For example, the Hardware category pertains to the needs of connecting devices with one another, which might be an Arduino Mega controller with an Android device or enabling. These needs evolve over time and so do the topics associated with each category. We study this evolution to record and help the growing and changing IoT community and to identify any gaps that still need attention.

4.2.2 Approach

Studies exist that identified popular IoT topics, in particular for the IoT hardware and hardware architectures for IoT applications [48]. Whitmore et al. [112] also studied the evolution of IoT topics, specifically in the contexts of healthcare and supply-chain. Surveys exits discussing protocols, technologies, and applications, with a focus on the problems reported by academia for specific IoT applications [112] or on a social-science perspective [48].

In this research question, we consider our four main categories of IoT topics from the developers' perspectives, as reported on the professional developers' Q&A Web site Stack Overflow, and we study the absolute and relative impacts of the topics identified in each category as follows.

Step 1. Compute Topic Absolute Impact. We apply topic popularity metrics as proposed in a previous work [39] to compute the popularity of a topic z_k within corpus c_j for a post d_i, where i can be any topic within corpus c_j. Formally, the popularity of each topic is defined as:

$$popularity(z_k, c_j) = \frac{|d_i|}{|c_j|} : dominant(d_i) = z_k, 1 \leq i \leq c_j, 1 \leq j \leq K \tag{3}$$

We apply LDA for corpus c_j to get a set of K topics (z_1,.....,z_k). We denote the probability for a specific topic z_k in a post d_i as $\theta(d_i,z_k)$ to define the absolute impact metric of a topic z_k in a month m as:

$$impact_{absolute}(z_k; m) = \sum_{d_i=1}^{D(m)} \theta(d_i; z_k) \tag{4}$$

where $D(m)$ is the total number of posts in month m. We further refine this absolute-impact metric for a category C for a specific month as:

$$impact_{absolute}(C;m) = \sum_{z_k}^{C} impact_{absolute}(z_k;m) \tag{5}$$

The category C belongs to four major category of IoT topic i.e., *Hardware, Software, Network and Tutorial.*

Step 2. Compute Topic Relative Impact. We use the relative impact metric to calculate the relative impact of IoT topic in a specific time, inspired from a previous work [39]. We define the relative impact metric of a topic z_k in m as:

$$impact_{relative}(z_k,m) = \frac{1}{|D(m)|} \sum_{d_i=1}^{\theta} (d_i;z_k), 1 \leq i \leq c_i \tag{6}$$

where $D(m)$ is the total number of posts in month m that contain topic z_k. The θ shows the probability of particular topic z_k for a post d_i. The relative impact metric estimates the proportion of posts for a specific topic z_k relative to all posts in a particular month m. We also apply the relative impact metric on categories:

$$impact_{relative}(C;m) = \sum_{z_k}^{C} impact_{relative}(z_k;m) \tag{7}$$

where C is the set of posts related to one of the four major categories of IoT posts.

4.2.3 Results

Using the previous equations, we calculated the impact of specific IoT topics and categories between year 2008 to 2019.

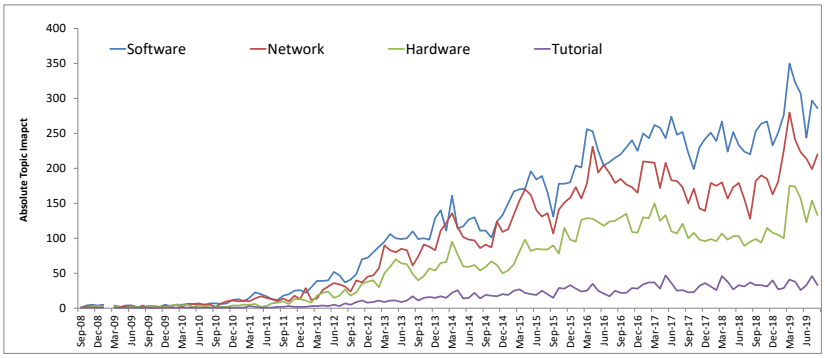

Fig. 5: The absolute impact scores of the topic popularity

Topic Absolute Impact. We explore the trends for the absolute impact of the IoT topics for the four Software, Network, Hardware, and Tutorials categories from September 2008 to September 2019, which is shown on Figure 5.

We observe that a trend starts from September 2008 and gradually increases for the Software category when compared to the Network category. Hardware attracted attention in May 2012. The number of posts for Network increased between September 2011 (32) to January 2012 (38) and May 2012 to (64). The significant increase in absolute topic impact for Software and Network indicates the growth of the IoT topics on SO, without any inflexion point until September 2019. Among the topics of various categories, Software- and Network-related topics intersect between January 2014 and May 2014, and also from May 2016 to September 2016.

We further study the most popular topic in the Software and Network categories, especially where their trends intersect, i.e., January 2014, September 2014, and May 2016. The most popular topic for Software in January 2014 was Android and different Android-related topics, e.g., playing repeated audio tracks in Android activity in $Q_{20889627}$. Topics related to Raspberry PI were also gaining in popularity, e.g., how to fix a segmentation fault using Alsa on a Raspberry PI in $Q_{20898454}$. We also observe similar topics associated in Network for Arduino and data transfer, e.g., getting data from serial device using Arduino in $Q_{24487480}$.

Similarly, in September 2014, popular topics mostly related to the programming of Arduino, e.g., what happens if the interrupt occurs while ISR is running in $Q_{25927694}$, and Raspberry PI, e.g., multiple vs. multipurpose sockets in $Q_{25952216}$). We also observe topics related to IoT open-source applications. We also observe a trend towards the use of some programming languages, e.g., node.js specifically targeting memory issues raised or run-time error associate to Raspberry PI in $Q_{37318124}$ and $Q_{37315548}$.

Interestingly, the absolute impact of the Hardware category slightly increases in September 2104, mainly due to issues with micro-controllers, e.g., how to make Mac detect AVR board using USBasp and burn program to it in $Q_{25591406}$, and programming needs for hardware devices, especially for the Raspberry PI, e.g., camera auto-capture using Python in $Q_{25592240}$).

The Tutorials category does not increase in popularity much, although we observe a slight increase after 2017 thanks to open-source IoT architectures, techniques, and tools and discussion of their common features, e.g., Gradle build tool in $Q_{44098797}$ and $Q_{43824128}$).

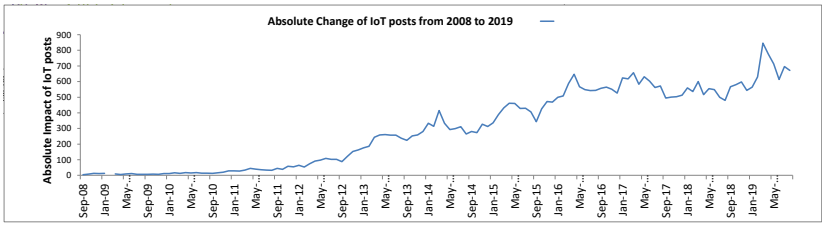

Fig. 6: Total absolute impact of the topic popularity

Figure 6 shows the trend in absolute popularity of IoT topics with peaks in mid of the years 2012 to 2019. We also observe similar trends with the categories Software and Network. The most popular topics in the IoT community are related to the Software category, Arduino, and Raspberry PI. Software and Network posts

were popular to discuss issues related to Arduino. We also observe a slight increase in 2014 and 2017 for Hardware while Tutorials did not gain much attention.

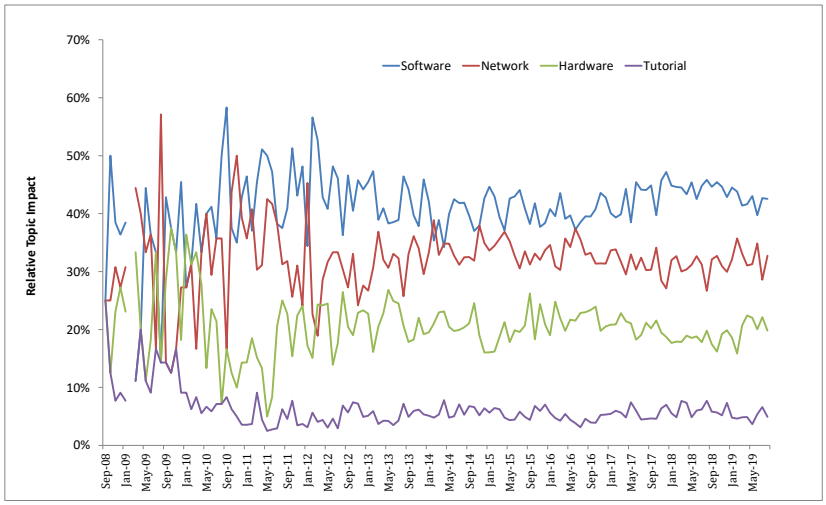

Fig. 7: The relative impact scores of the topic popularity

Topic Relative Impact. We compute the relative impact using Equation 6 for each of the topics for the four categories Software, Network, Hardware, and Tutorial. Figure 7 shows the relative change in popularity of the topics, which indicates the distribution for each topic.

We observe an overall increase for Software-related topics beginning from September 2008 to 2019. Network- and Hardware-related topics also get attention from 2008 with high relative change from May 2009 to September 2009. There is an interesting trend from May to September 2009 with the intersection of Software, Network, and Hardware.

Since June 2009, the Software category discusses more programming problems associated with the .NET Framework, e.g., listening to serial COM ports in Q_{915904}, and run-time errors for Arduino in $Q_{1013936}$. We also observe a trend of issues related to communications with the Remote Audio Data system, e.g., RAD and BlinkM in $Q_{1468966}$, how to control a BlinkM with an Arduino through RAD in $Q_{1468966}$.

Network-related topics focus on similar issues with (1) connecting devices to Arduino, e.g., interfacing a DF Robot Bluetooth module with Arduino in $Q_{1366927}$, (2) hardware Programming, e.g., how to learn hardware programming in $Q_{1252428}$, and (3) creating network adapters, e.g., $Q_{1017005}$. The posts associated to the category Hardware also discuss issues regarding specific chips and boards, in particular Arduino, e.g., the difference/relationship between AVR and Arduino in $Q_{1447502}$, specific programming practices in $Q_{1013936}$, and their uses in Q_{949890}.

The most popular topics in the IoT community in the Software category are mainly associated to C/C++ and the .NET Framework. There is also intersections of the Software and Network categories in 2009, 2014 and 2017. Software and Network mostly discuss issues related to Arduino and communication modes for specific versions. We also observe a slight increase in 2014 and 2017 for Hardware.

We observe a constant increase for the Software, Network, and Hardware categories from 2015 to 2016, related to connecting IoT devices with Arduino and some operating systems, e.g., Android App (Kivy or Ai) freezes when connecting BLE on Arduino Uno in $Q_{34394148}$, and issues with specific feature, e.g., Arduino PID DC motor position control in $Q_{43818818}$ and $Q_{51030657}$. We also observe discussions associated to the use of programming languages for particular devices, e.g., printing UTF-8 multi-byte characters on Raspbian in $Q_{28340275}$.

Summary of RQ2: How do the IoT topics evolve over time? The discussions are mainly associated to communication problems related to Software, Network, and Hardware related topics - by using popular devices like Auduino and Raspberry PI and popular programming languages like C/C++ and .NET Framework. This trend increases from 2009, with the use of programming languages for open-Source systems gaining in popularity. We also observed that using Arduino for connectivity between software and hardware is a popular topic.

4.3 What Types of Questions Do IoT Developers Ask about IoT Topics? (RQ3)

4.3.1 Motivation

After examining the most popular topics of discussions on the IoT, we analyse the types of posts in each category to identify the issues and challenges faced by developers. This analysis allows proposing enhancements to solve/overcome these issues/challenges. Given IoT is a new, emerging paradigm, we want to understand what types of questions developers are asking, e.g., asking for solutions (How) or for clarification (What/Why), or both. This information offers insight into the type of support that IoT developers need. For example, if most of the questions in a topic is of type How, developers need better documentation/tutorials and official learning resources are lacking. If developers ask many What questions, they need guidelines to choose IoT architectures, techniques, and tools or to assess the requirements for their IoT projects.

4.3.2 Approach

To understand the intentions of the different types of posts, we collect a statistically significant sample of all questions in our dataset and then manually analyze each question and labe the question into one of four types: How, Why, What, and Others. The four question types were originally used to categorize chatbot questions SO by Abdellatif et al. [1].

Step 1. Generate Sample. Our dataset has total 39,305 questions. A statistically significant sample with a 95% confidence level and 5 confidence interval would require at least 380 random questions. A random sample from the entire 39,305

questions will give us a sample representative of the entire dataset (i.e., questions). As such, a random sample could miss questions from a subset of questions that may belong to specific topic category, where the subset size can be very small compared to the entire dataset. As noted in Section 4.1, the 40 topics that we observed in the 39,305 questions can be grouped into four categories: Software, Hardware, Network, and Tutorial. Given that the distribution of questions per category is not uniform, a random sample of 380 questions might miss many important questions from categories with fewer questions (e.g., Tutorial). Therefore, following Abdellatif et al. [1], we draw a statistically significant random sample from each of the four categories. With a 95% confidence level (and 5 interval), the distribution of questions in the samples for the four categories are as follows: 375 Software posts (from a total of 162,302), 373 Network posts (from a total of 13,044) posts, 366 Hardware posts (from a total of 7,905), and 325 Tutorial posts (from a total of 2,092). Overall, we have sampled and manually analyzed a total of 1,439 questions from our entire 39K questions.

Step 2. Label Question Types. We label each post from our samples using previous categories [84], which we adapted to the IoT:

- How: posts that raise questions about the use/implementation of architectures, techniques, or tools. This category of posts focuses on the steps required to achieve specific goals, e.g., *" Message between bash and javascript via named pipes?"* in $Q_{17828014}$. Questions about bug fixing are also included in this category.
- Why: posts reporting or discussing the reason, cause, or purpose for a specific behaviour. These posts are mostly related to troubleshooting. They help developers understand or explain a problem-solving approach, e.g., *"Python mechanize on Raspbian?"* in $Q_{17503447}$.
- What: posts seeking information about a particular architecture, technique, or tool as well as problem or event. These posts pertain to issues associated with code crash, run-time errors, memory management, particular framework or device. They provide information helping developers make informed decisions, e.g., *"Arduino ADK ways to connect to an Android tablet"* in $Q_{16611421}$.
- Other: posts that do not fall into any of the three previous categories, e.g., *"Particle photon johnny-five particle-io interfacing"* in $Q_{35752088}$.

Each post was labeled by the first and before-last authors. We assessed their level of agreement using Cohen's Kappa [58]. In general, they achieved a substantial agreement ($\kappa = 0.80$) on the 1,439 classified posts. This level of agreement is on par with those reported in previous work [1, 84]. The few cases of disagreement were resolved through discussions by revisiting the questions, discussing each other's views until a consensual decision was reached, as in previous work [84]. We apply three iterations to identify the correct label for each question type. The data of revisiting the questions, discussion and final agreement decision regarding question type is also available online.

Many posts have multiple labels, e.g., How and What. The title of the post may be different from the intention of its content. Posts label as Why also tend to be labeled How, e.g., *"Class in parameter of function (Arduino) does not compile"* in $Q_{18304453}$. Hence, the sums of the percentages reported below account for more than the 1,439 classified posts.

4.3.3 Results

Table 1: Types of questions across the IoT topic categories

Topic Category	How	What	Why	Other
Software	41.6%	43.5%	26.4%	4.0%
Network	51.5%	39.1%	17.4%	0.4%
Hardware	58.0%	36.3%	19.1%	3.0%
Tutorial	38.2%	32.6%	15.1%	28.3%
Overall	47.32%	37.87%	19.59%	8.34%

Table 1 shows the percentages of each type of questions for the four high-level IoT categories.

How: This type of questions is predominant in the categories Hardware (58%), Network (51.5%), and Software (41.6%), e.g., $Q_{18806141}$, and account for 32.16% in Tutorials, e.g., $Q_{5931572}$. Such posts in Hardware and Network pertain to common, recurring problems related to protocols, network troubleshooting, specifically in relation to low-level and middle-level devices, e.g., $Q_{27439367}$. They also relate to technical issues, e.g., $Q_{33756224}$, memory management, e.g., $Q_{27744747}$, or specification issues, e.g., $Q_{9805829}$. Open-source IoT operating systems are one of the most discussed topic in the posts, e.g., $Q_{13781908}$.

What: This type of questions is predominant in the Software category (43.5%), e.g., $Q_{17934385}$ or $Q_{20752741}$, in which developers are often interested in solving specific problems, e.g., file handling errors in $Q_{17934385}$, (2) compile- and run-time errors, e.g., $Q_{20752741}$, and (3) errors associated to certain practices, e.g., Software as a Service (SaaS) in $Q_{26439006}$.

Why: This type of questions happens in Software (26.4%), mostly associated to using Arduino, e.g., Arduino can't get my boolean to work in $Q_{27242253}$, in Hardware (19.1%), and in Network (17.4%). They include questions about compilation errors, e.g., *"Why I cannot use the struct as a type on my function parameter using C/C++ for Arduino compiler design"* in $Q_{29201740}$; errors with specific hardware–software combinations, e.g., *"Using go.dbus with omxplayer on Raspberry Pi"* in $Q_{28030045}$ and *"getting error on Supervison on supervisorctl ERROR (no such process)"* in $Q_{28145360}$. Most of the Why posts are associated to Raspberry PI, e.g., $Q_{29198312}$, $Q_{28145360}$, and Arduino, e.g., $Q_{29201740}$, $Q_{27242253}$.

Other : This type of questions is most prevalent in the Tutorial category with 38.1%, e.g., $Q_{13952519}$. The posts are about general problems, e.g., *"mono 3.0.1 asp.net An assembly with the same identity 'mscorlib has already been imported"* in $Q_{13672672}$. Consider removing one of the references, e.g., *"json data returns invalid label error"* in $Q_{3672672}$ or *"Trying to build "Hello, world!" media player activity using Jelly Beans new low-level media API"* in $Q_{13387707}$. These posts also include posts with unsure/multiple questions, e.g., how to move a head?, is this a bug?, did the installation go wrong?, can I fix this?

Summary of RQ3. What types of questions do IoT developers ask about IoT topics? Hardware and Network posts have the greatest numbers of questions of type How compared to Software and Tutorials. The type of questions Other mostly belong in Tutorials, discussing options, criteria, usage of software and hardware.

4.4 How Do the Popularity and Difficulty of the Topics Vary? (RQ4)

4.4.1 Motivation

Our findings from RQ1 show that there are diverse types of IoT topics discussed in SO. Our findings from RQ3 show that many of the questions are of types How and What. Thus, IoT developers seem to have difficulty with certain architectures, techniques, and tools (i.e., How) as well as in understanding their functioning (i.e., What). Despite these difficulties, some topics are recurring in the posts. Thus, all topics are not equally popular and difficult. A study of the popularity and difficulty of the topics offers insights into prioritizing research and developers' efforts. For example, newcomers to the IoT could focus on more popular topics. IoT researchers could devise ways to make some architectures, techniques, and tools more usable.

4.4.2 Approach

We compute three metrics to measure popularity for each topic: (1) Average number of views for all questions assigned to the topic, (2) Average number of questions of a topic marked as users' favorite, and (3) Average score of questions of a topic. The three measures (i.e., view count, score, favorite count) are standard features for a question in SO, i.e., they were introduced by SO team to analyze the popularity of a question. We compute two metrics to measure the difficulty of getting answers for each topic: (1) Percentage of questions that have no accepted answers, and (2) Average median time needed for a question of a topic to receive accepted answer. In SO, one of the answers to a question can be marked as accepted by the asker of the question. Given that the asker of the question explicitly provides his/her feedback by accepting an answer, the accepted answer to a question is perceived correct and-or good quality. As such, the absence of an accepted answer may denote that the asker did not find an answer that can be accepted. While the lack of quality of a question can be problematic to get an answer, the SO community collaboratively edits posts to improve question/answer quality. As such, the lack of an accepted answer may most likely denote that the question may be perceived as difficult by other developers to provide an answer. The success of a crowd-sourced platform like SO depends largely on the developers to provide quick and correct answers. The median time to get an answer to a question is only 21 minutes, but a difficult question might need more time to receive an accepted answer [92]. The above five metrics were also previously used in several research papers to compute the popularity and difficulty of topics found in SO posts [1, 3, 13, 82, 119], e.g., by Bagherzadeh and Khatchadourian [13] to study big data topics, by Ahmed et al. [3] to study concurrency topics, by Abdellatif et al. [1] to analyze chatbot topics, etc.

Having multiple metrics to measure a characteristic can create confusion if the ranking from a metric differs from the ranking of another metric. This can

occur for our topic popularity or difficulty analysis because each characteristic has more than one metric. We, therefore, create two fused metrics, one to measure the popularity of a topic and another to measure the difficulty of the topics. We describe the two metrics below.

Fused Popularity Metric. We first compute the three popularity metrics for each topic. The average view counts of a topic can be in range of thousands, average scores between 0-2, and average favorite between 0-3. Therefore, we normalize a metric value of a given topic by dividing the metric value by the average of the metric values across all the 40 topics. Thus, we create three new variables, one each for the three normalized metric values. Suppose the normalized metrics of a given topic i are called $ViewN_i$, $FavoriteN_i$, $ScoreN_i$.

$$ViewN_i = \frac{View_i \times 40}{\sum_{j=1}^{40} View_j} \tag{8}$$

$$FavoriteN_i = \frac{Favorite_i \times 40}{\sum_{j=1}^{40} Favorite_j} \tag{9}$$

$$ScoreN_i = \frac{Score_i \times 40}{\sum_{j=1}^{40} Score_j} \tag{10}$$

We calculate the fused popularity $FusedP_i$ of topic i by taking the average of the above three normalized metric values.

$$FusedP_i = \frac{ViewN_i + FavoriteN_i + ScoreN_i}{3} \tag{11}$$

Fused Difficulty Metric. We first compute the two difficulty metrics for each topic. Similar to the popularity metrics, we normalize a metric value of a given topic by dividing the metric value by the average of the metric values across all the 40 topics. Thus, we create two new variables, one each for the two normalized metric values. Suppose the normalized metrics of a given topic i are called $PctQWoAcceptedAnswerN_i$, $MedHrsToGetAccAnsN_i$.

$$PctQWoAcceptedAnswerN_i = \frac{PctQWoAcceptedAnswer_i \times 40}{\sum_{j=1}^{40} PctQWoAcceptedAnswer_j} \tag{12}$$

$$MedHrsToGetAccAnsN_i = \frac{MedHrsToGetAccAns_i \times 40}{\sum_{j=1}^{40} MedHrsToGetAccAns_j} \tag{13}$$

We calculate the fused difficulty metric $FusedD_i$ of topic i by taking the average of the above two normalized metric values.

$$FusedD_i = \frac{PctQWoAcceptedAnswerN_i + MedHrsToGetAccAnsN_i}{2} \tag{14}$$

We also assess the correlation between each of three topic popularity and two difficulty metrics. We use Kendall's τ correlation measure [45]. Unlike Mann-Whitney correlation [47], Kendall's τ is not susceptible to outliers in the data. We cannot compute the evolution of the popularity and difficulty metrics, because SO data does not provide the basic data over time. However, given that all of the SO topics are showing trends of increasing in recent years, as shown in RQ2, our analysis of topic popularity and difficulty is valid for recent posts.

Table 2: Popularity of IoT topics

Topic	Category	FusedP	#View	#Favorite	#Score
Microcontroller Configuration	Hardware	1.46	2361.5	1.8	1.1
Serial port communication	Network	1.44	2356.8	1.8	1.0
Data Parsing	Software	1.43	2702.9	1.6	0.9
BLE	Network	1.41	1550.6	2.1	1.3
General Troubleshoot	Software	1.37	1719.4	1.9	1.2
Audio Processing	Software	1.31	1225.8	1.7	1.4
Installation Tutorial	Tutorial	1.30	1790.5	1.7	1.1
General IoT Tutorial	Tutorial	1.29	1490.1	2.4	0.9
Linux Interfacing	Software	1.28	1730.7	1.6	1.1
Multithreading	Software	1.28	1252.5	2.0	1.2
Build Troubleshoot	Software	1.21	1332.7	2.0	1.0
Device to Internet	Network	1.16	1458.6	1.9	0.8
Variable Debugging	Software	1.14	2062.5	1.4	0.7
Memory management	Hardware	1.14	1499.2	1.6	0.9
Container Management	Software	1.10	1087.2	1.6	1.1
Wireless Networking	Network	1.08	1422.9	1.7	0.8
Windows IoT	Software	1.03	996.2	1.4	1.0
Graphics/Touchscreen	Hardware	1.02	1455.6	1.4	0.8
Multimedia Streaming	Software	0.98	1201.5	1.4	0.8
Exception Handling	Software	0.96	1055.3	1.4	0.9
Connection Debugging	Network	0.94	1171.7	1.4	0.7
GPIO Troubleshoot	Network	0.93	1190.3	1.3	0.8
Core OS/SDK	Software	0.93	990.7	1.5	0.8
Device Troubleshoot	Hardware	0.91	1532.9	1.1	0.6
Library Troubleshoot	Software	0.89	1428.7	1.1	0.7
Time zone/formatting	Software	0.87	1239.4	1.2	0.7
Secure messaging	Network	0.86	978.5	1.4	0.7
Script Scheduling	Software	0.85	1150.2	1.3	0.6
I/O Troubleshoot	Software	0.84	1348.2	1.2	0.5
HTTP request handling	Network	0.79	1024.4	1.3	0.5
D2D Communication	Network	0.77	1095.8	1.1	0.6
Python IoT APIs	Software	0.76	1116.5	0.9	0.6
ESP8266/Wifi-Microchip	Hardware	0.74	1092.9	1.2	0.5
Communication Troubleshoot	Network	0.71	832.8	1.2	0.6
LED Configuration	Hardware	0.71	1019.3	1.1	0.5
GPS Coordinates & Positions	Network	0.69	936.0	1.2	0.4
Sensor Feeds	Hardware	0.68	1014.4	1.2	0.4
Signal Troubleshoot	Hardware	0.61	760.7	1.0	0.5
Performance Debugging	Software	0.57	803.1	0.7	0.5
IoT Hub	Software	0.54	334.6	1.1	0.5

4.4.3 Results

We first discuss topic popularity. We then explore topic difficulty. Finally, we discuss the correlation between topic popularity and difficulty.

Topic Popularity. Table 2 shows four popularity metrics for each IoT topic: average number of 1. view counts, 2. favorite counts, 3. scores, 4. answers per questions under the topic, and 5. The overall popularity of a topic based on the linear fusion of the above three metrics using Equation 11. Topics are ranked by the FuseP column in descending order, i.e., the fused popularity metric.

Microcontroller configuration topic from the Hardware category has the highest FuesP value. This topic contains discussions about the configuration and use of microcontrollers, sensors, and memory management in IoT. This topic The Serial

Table 3: Difficulty of getting answers across IoT topics

Topic	Category	FusedD	Hrs To Acc.	W/o Acc. Ans
IoT Hub	Software	1.39	7.0	60%
Windows IoT	Software	1.38	6.8	64%
Linux Interfacing	Software	1.37	6.8	65%
ESP8266/Wifi-Microchip	Hardware	1.29	6.3	69%
BLE	Network	1.09	5.0	73%
Multimedia Streaming	Software	1.06	4.9	74%
Secure messaging	Network	1.06	5.0	66%
Core OS/SDK	Software	0.89	4.0	69%
Installation Tutorial	Tutorial	0.85	3.7	69%
Audio Processing	Software	0.85	3.8	66%
GPIO Troubleshoot	Network	0.85	3.8	63%
Wireless Networking	Network	0.83	3.6	68%
Exception Handling	Software	0.83	3.8	57%
GPS Coordinates & Positions	Network	0.79	3.4	68%
Connection Debugging	Network	0.76	3.2	69%
Container Management	Software	0.76	3.2	70%
Microcontroller Configuration	Hardware	0.75	3.2	64%
Graphics/Touchscreen	Hardware	0.73	3.0	69%
HTTP request handling	Network	0.70	3.0	65%
Serial port communication	Network	0.66	2.8	64%
Device to Internet	Network	0.59	2.3	67%
Time zone/formatting	Software	0.58	2.2	64%
Library Troubleshoot	Software	0.56	2.1	67%
Sensor Feeds	Hardware	0.53	1.9	70%
D2D Communication	Network	0.52	2.0	61%
Multithreading	Software	0.52	2.0	59%
Device Troubleshoot	Hardware	0.49	1.7	67%
LED Configuration	Hardware	0.47	1.6	65%
Build Troubleshoot	Software	0.42	1.5	56%
Performance Debugging	Software	0.41	1.3	64%
Signal Troubleshoot	Hardware	0.40	1.2	67%
I/O Troubleshoot	Software	0.37	1.1	64%
Script Scheduling	Software	0.36	1.0	63%
Memory management	Hardware	0.35	1.1	55%
Variable Debugging	Software	0.34	1.1	52%
Python IoT APIs	Software	0.34	0.9	64%
General IoT Tutorial	Tutorial	0.33	1.0	56%
Communication Troubleshoot	Network	0.30	0.9	51%
General Troubleshoot	Software	0.28	0.7	58%
Data Parsing	Software	0.27	0.7	55%

Port Communication topics from the Network category has the second highest FuseP value. This topic contains discussions about connections between IoT devices by using the serial ports. The Data Parsing topic in Software is the third most popular topic in terms of FuseP value. This topic has the highest number of views and the greatest average number of answers per question. The posts under this topic discuss about the parsing of data from different sources. For example, $Q_{36804794}$ asks about *"iterparse large XML using python"* in Raspberry PI 2. The OP explains that *"This has been driving me nuts all day and i would appreciate a bit of help with parsing a large XML file"*. Generally, data parsing questions are due to the limited memory and specific communication interfaces of IoT devices.

General IoT Tutorial in Tutorials has the highest number of favorite posts. The posts are specifically about learning material. For example, $Q_{44114645}$ asks

"simple language source code for Arduino". The OP explains: *"I'm an elementary school teacher. Next year I want to teach my class a bit about hardware and software as extracurricular lessons. For these lessons, I started a new project in Arduino"*.

The IoT Hub topic from the Software category is the least popular, with only 3% of all questions and a FuseP value of 0.54 compared to 1.46 FuseP of the most popular topic (i.e., Microcontroller Config). Many questions in the IoT Hub topic are about Azure IoT Hub, which provides a cloud-hosted middleware to connect IoT devices. Many of the questions remain unanswered. Previous research acknowledged the challenges to develop middleware solutions for the IoT [26].

Topic Difficulty. Table 3 presents the three difficulty metrics per topic: 1. Percentages of questions without an accepted answer, 2. Median hours taken to get an accepted answer, and 3. FuseD value per topic based on the above two metric values. The first two metrics measure the difficulty of getting a corrected answer to a question. The topics are ranked by FuseD value in descending order.

The two cloud-based/OS-based topics (IoT Hub and Windows IoT) are ranked as the most difficult. The third topic 'Linux Interfacing' is about the usage of Linux for IoT development. Thus, topics related to the IoT development are the most difficult to get accepted answers in SO. If we look at the topic popularity values in Table 2, the most difficult topic, IoT Hub, is ranked as the least popular Table 2 based on the FuseP metric value. Windows IoT topic, while ranking as the second most difficult, is situated in the top half of popularity values in Table 2,which shows that IoT developers are interested to use the IoT solutions based on Microsoft Windows, but they do not have enough support in SO to get correct answers. The Miscorosft IE Edge team has recently moved their entire support of Q&A to SO. Perhaps, they can take similar actions to support Windows IoT developers in SO.

The Data Parsing topic from the Software category is the least difficult in terms of FuseD value. Questions related to this topic are viewed by many and have higher proportions of accepted answers than other topics. While Data Parsing in Software is the most popular topic in terms of average views, Multimedia Streaming in the same category is the most difficult based on the percentages of questions without accepted answers (74%) as well as the average time to get an accepted answer (8.2 hours). Many questions in this topic have as many as seven answers, yet none marked as accepted. For example,$Q_{23538522}$ is about *"Scanning QR Code via zbar and Raspicam modul"*.The question was asked five years ago and last edited in November 2019, which shows that the problem is still relevant. It has been viewed more than 25,000 times. It has seven answers, yet none is accepted as the correct answer. Similarly, $Q_{38302161}$ states that "cv2.videocapture doesn't works on Raspberry-pi" . This question was asked four years ago and has been viewed 4,000 times. It has five answers, with the most recent answer in October 2019. However, none of the answers is accepted.

Among the topics in the Hardware category, the ESP8266/Wifi-Microchip topic is ranked as the most difficult because around 69% of its questions remain without accepted answers, while those that get an accepted answer normally had to wait a median of 5 hours. Among the topics in the Network category, BLE topic is the most difficult. Questions under this topic are related to the usage, connection, and positioning of BLE. Similar to the Multimedia Streaming topic, more than 70% of questions in BLE are without accepted answers. For example, $Q_{27059556}$ reports

that the OP "Cant connect the HM-10 bluetooth to Arduino Uno" . It was asked 5.5 years ago, viewed more than 14,000 times, but has no accepted answer.

Correlation between Topic Popularity and Difficulty. Our results from topic popularity and difficulty indicate that there could be an inverse relationship between the popularity and difficulty of a topic. For example, the topic Data Parsing is most popular but also least difficult, as shown in Tables 2 and 3. This observation is less clear for other topics, such as BLE, which is the second most difficult, yet only the eighth most popular.

Table 4: Correlation between the popularity and difficulty

coefficient/p-value	**View**	**Favorites**	**Score**
% w/o acc. answer	-0.10/0.35	-0.03/0.78	-0.05/0.66
Median Hrs to acc. answer	-0.12/0.28	0.10/0.35	0.18/0.10

Table 4 shows nine correlation measures between the difficulty and popularity metrics in Tables 2 and 3. Seven out of nine of the correlation coefficients are negative, which confirms our hypothesis that the popularity of topic decreases with an increased difficulty. Yet, correlation measures are not statistically significant at a 95% confidence level. Nonetheless, IoT educators could use this insight to produce viable and acceptable solutions to difficult questions and promote certain topics to make them more popular.

Summary of RQ4. How do the popularity and difficulty of the topics vary? While one topic, Data Parsing in the Software Category, is the most popular in terms of page views; another topic, Multimedia Streaming in the same category, is the most difficult in terms getting an accepted answer. Three OS/cloud-based topics from the Software category are ranked as the most difficult in terms of the fused difficulty metric. The BLE topic in the Network category is the fifth most difficult topic (first among all Network topics), yet it is fourth most popular topic, showing the need for better tutorials. The difficulty and popularity metrics are negatively correlated in five out of the seven correlation measures, i.e., more difficult topics are generally less popular. For example, the IoT Hub from Software category is the most difficult and the least popular topic, while Data Parsing from Software category is the least difficult and third most popular topic.

5 Discussions

In this section, we first investigate the stability of the produced topics in Section 5.1. We then explain why we do not consider 'not accepted' answers in our topic modeling in Section 5.2. Finally, we compare our study findings with similar, previous works that used topic modeling on SO posts, in Section 5.3.

5.1 Topic Stability Analysis

We use the widely-used LDA algorithm to find topics in IoT developer discussions in SO. LDA uses Dirichlet distribution, which is a multivariate distribution, and a set of parameters to identify topics, like K as optimal number of topics,, α as document-topic density, and β as topic-word density. As Agrawal et al. [2] observed, if we do not tune the parameters, the identified topics may change in multiple runs of LDA on the same dataset. As we noted in Section 3, we followed standard topic coherence measurement technique to determine optimal number of topics in our dataset. We also followed recommendations from literature to determine document-topic density (α) and topic-word density (β) values. Nevertheless, we must ensure that the reported IoT topics are stable between two different runs of LDA on our IoT dataset and we apply the following qualitative and quantitative steps to systematically determine the stability of the IoT topics in our dataset:

1. We run LDA on our SO IoT dataset again with the same parameters used to produce the topics in Section 4. We denote the topics in this new run as 'R2' Topics and the topics reported in Section 4 as 'R1' topics.
2. We manually analyze each of 50 topics from R2 and assign a label to each topic, using approach similar to Section 4.1.2. This means that we manually analyzed 15-30 questions that are assigned to a topic. Some of the questions are picked randomly and some are picked by sorting the questions based on their correlation with the topic, i.e., these questions have the highest correlation score with this topic among all topics.
3. We revisit the manual labels of each topic several times and merge topics that contain similar questions and answers. For example, we label the topic ID 40 in R2 as 'BLE and Core OS/SDK', because the questions contain discussions about Bluetooth Low Energy (BLE) devices as well as IoT SDKs that can be used to interface with BLE devises and other core features.
4. We compare the final list of topics between R1 and R2.

In Table 5, we show the topics in R1 and R2 after the manual labeling. We find the same 40 topics in both R1 and R2. Out of the 40 topics, we observe a one-to-one mapping for 39 topics. The column 'Joint Match' in Table 6 shows that one topic in R2 was merged with another topic: 'BLE and Core OS/SDK'. Upon close observation of the two topics in R1, we report that the two topics could have been merged in R1 as in R2. Besides the merged topics, we do not find any topics missing between R1 and R2: there are no new topics in R2, which confirms that the parameters provide the optimal number of topics for our dataset. In R2, as we noted, two topics from R1 are found together: 'BLE and Core OS/SDK'. We checked the reason for the two topics found together in R2. In R2, we see questions related to 'Core OS/SDK' features in ID 40 as follows: (1) *"Android Things send GPS data to TextView"* ($Q_{50932499}$), where the developer attempts to display GPS information on screen using the Android Things OS. (2) *"Android Things and RXTX library"* ($Q_{53160016}$), where the developer was finding it difficult use the RXTX (a Java library for serial and parallel communication) in the Android Things OS. Both are among the in the top 10 questions that are associated to the topic (based on topic-documentation correlation score). We also find questions related to Bluetooth low energy devices (BLE) in the top 10 questions as: (1) *"Use Bluetooth to control Arduino from IOS app"* ($Q_{57337168}$), where the developer wants

to create his own iOS app to control his Arduino device by communicating with the Ardunio via the BLE module in the Arduino device. (2) *"Swift 3 arduino Uno HM-10 Ble - Notifications on iphone"* ($Q_{44250540}$), where the developer wants to receive notifications in his iphone from his Arduino Uno device via the BLE interface and he wants to use the Swift programming language libraries. Therefore, in R2 LDA considered both topics as similar due to the presence of keywords like libraries and OS in the questions of both topics.

Table 5: Matching of Topics (T) between the two runs (R1, R2) of LDA

R1#T	R2#T2	Exact Match	Joint Match	Missed from R1	New in R2
40	40	39	1	0	0

The manual validation of the topics between R1 an R2 offer us confidence in the stability of topics from Section 4. However, a manual analysis can always be subject to unconscious subjective bias. Therefore, we investigate five algorithms to automatically match a topic in R2 with a topic in R1. For each algorithm, we first pick a topic ID i in R2 and compare it against all topics in R1 based on a matching condition, with i and j between 0 and 49:

1. **Max Similar All Q (Q all).** We assign i from R2 to topic ID j from R1, if i and j have the maximum number of common questions among all the topic IDs (i.e., IDs 0 to 49) in R1.
2. **Max Similar All Q+A (Q+A all).** We assign i from R2 to topic ID j from R1, if i and j have the maximum number of common questions and accepted answers among all the topic IDs (i.e., IDs 0 to 49) in R1.
3. **Max Similar Q in Top 100 Q (Q T100).** We assign i from R2 to topic ID j from R1, if i and j have the maximum number of common questions among the top 100 questions between i and j. Top rank is determined based on topic to question correlation score.
4. **Max Similar Words in Top 30 Words (W T30).** We assign i from R2 to topic ID j from R1, if i and j have the maximum number of common questions among the top 30 words between i and j. Top rank is determined based on topic to words correlation score.
5. **Max Similar Words in Top 10 Words (W T10).** We assign i from R2 to topic ID j from R1, if i and j have the maximum number of common questions among the top 10 words between i and j. Top rank is determined based on topic to words correlation score.

For a given matching condition and for a given topic ID i from R2, if we find more than one topic ID in R2 with the maximum similarity score, we assign i to all of those topic IDs in R1. Once we finish the assignment of topics between R1 and R2 based on the above algorithms, we compare the assignments with our manual assignments. We check whether a suggested match by an algorithm agrees with the assignment from our manual labeling. In Table 6, we show the performance of the five algorithms by showing their percentage of agreement with the manual assignments.The maximum 95.5% agreement between manual and algorithmic assignment was achieved with the 'W T10' algorithm. In fact, we

Table 6: Percentage of Topics Matched by Each Matching Algorithm between R1 and R2 (Q = Question, A = Answer, W = Words)

Q All	**Q+A All**	**Q Top 100**	**W Top 30**	**W Top10**	**At Least One Matcher**
90.9%	90.9%	84.1%	88.6%	95.5%	95.5%

found at least 84% agreement between any algorithm and the manual assignment. This observation offers further evidence that the IoT topics are stable. We share all the data for both R1 and R2 in our online replication package.

5.2 The *Issues* with Not Accepted Answers

In this paper, we used questions and accepted answers in our topic modeling. We do not consider answers that are not marked as accepted. This decision is based on three observations. (1) Previous papers that used topic modeling in SO posts also only considered questions and accepted answers, e.g., big data topics in SO [12], concurrency topics [3], mobile app topics [83], chat bot topics [1], general technical topics in SO [15]. (2) A significant body of research involving SO posts finds that the quality of an answer in SO may not always be good and this is more prevalent for answers that are not accepted [9,72,80,98,116,121]. It is, therefore, a standard practice in SE research to not analyze answers to a question that are not marked as accepted. (3) An answer not marked as accepted may not be relevant to the question and there is no easy to determine its relevance automatically. Consider the example question and unaccepted answer in Figure 8. The question belongs to the topic 'BLE' in in our dataset. The question is about how to send notification to an iPhone via the Bluetooth interface of an Arduino device. There are two answers to this question. One question is marked as accepted by the asker (not shown in Figure 8). Another answer is not accepted (shown in Figure 8). The reason, as explained by the asker in the comment to the answer, is that the provided answer does not fully answer to the question and it is also not fully relevant to the problem described in the question. The answer also a score of -1, i.e., it is not considered helpful by the asker or other developers. As such, the inclusion of such answers to our IoT post analysis would have introduced noise and/or wrong insights about IoT discussions in SO. We thus decided not include unaccepted answers to a question in our analysis. We discuss missing of some potentially important insights such excluded answers in the threats to validity (Section 7).

5.3 IoT Topics Compared to Other Domains

As we noted in Section 2, SO posts have been the subject of several studies that used topic modeling to investigate topics for diverse domains like big data [13], chat-bots [1], blockchain [107], deep learning [39] and so on. Each study analyzed posts that contain discussions about a particular domain. The distribution and the nature of the questions differ across domains. Given that SO is arguably the most popular developer forum, such characteristics per domain may identify the

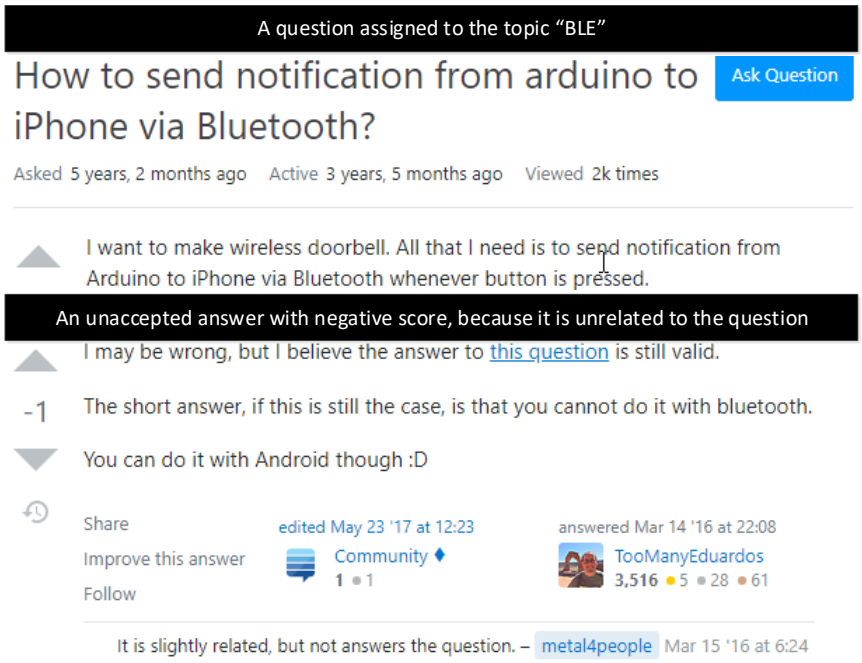

Fig. 8: An example question ($Q_{35998871}$) in our dataset with an unaccepted answer

state-of-the-practice tools and techniques per domain, as well as the level of engagement among developers. Therefore, a systematic comparison of the similarities and differences among domains is interesting. Therefore, we study all the different papers that used SO posts to study topics, no matter the domain. We specifically look for six metrics in the papers. Four out of the six metrics are related to the popularity of topics or the underlying domains: 1. Total number of posts analyzed in the study, 2. Average views, 3. Average favorite counts, 4. Average scores. The other two metrics are related to topic difficulty: 1. Percent of questions without an accepted answer, 2. Median hours to accept an answer per topic.

The purpose of this comparison is to report any similarities or differences of the characteristics of the IoT topics compared to that in other domains. We look at the metric values reported in the papers. We do not replicate the findings of each paper and do not preprocess any data from the papers. We only select a related paper for comparison if it reports the above metrics. Out of the related papers in the literature, we observed that the following five papers reported all the above metric for the domain of: big data [13], chat-bots [1], security [119], mobile apps [83], and concurrency [3].Although SO is also used for Blockchain s [107] and deep learning s [39], the two related papers did not report all the metrics.

Table 7 compares the seven metrics that we used in our study of IoT topics in SO with those used in previous studies for other domains: big data [13], chat-bots [1], security [119], and mobile apps [83]. There is a greater number of IoT

Table 7: Comparison of popularity and difficulty metrics between different domains (P = Popularity Metrics, D = Difficulty Metrics)

Type	Metrics	IoT	Big Data	Chatbot	Security	Mobile	Concurrency
P	# Posts	53,173	125,671	3,890	94,541	1,604,483	245,541
	Avg View	1,320.3	1,560.4	512.4	2,461.1	2,300.0	1,641
	Avg Favorite	1.5	1.9	1.6	3.8	2.8	0.8
	Avg Score	0.8	1.4	0.7	2.7	2.1	2.5
D	% W/o Acct Ans	64%	60.3%	67.7%	48.2%	52%	43.8%
	Med Hrs to Acc.	2.9	3.3	14.8	0.9	0.7	0.7

posts (questions + accepted answers) than chat-bot posts (53K vs. 3.8K) but it is less than that of the numbers for the other domains (big data, security, and mobile apps). As we noted above, SO data is also used to know the programmer discussion for Blockchain [107] and deep learning [39]. Total number of posts for Blockchain study [107] are 32,375 and for deep learning study [107] are 26,887. However, these two studies did not report the other metrics. These numbers show that the IoT is an emerging paradigm. As such, while the number of IoT-related discussions in SO may be lower than that of other domains, as we reported in RQ2, this number is rapidly increasing across all four IoT categories.

With respect to the other popularity metrics, IoT topics show numbers similar to big data for two metrics (Average View and Number of Answers) and similar to chat-bot for two other metrics (Average Favorite and Average Score). Overall, the popularity metric values for IoT topics are closer to those of big data and chat-bot than to those of the other two domains. We explain this observation by the fact that Big Data, Chat Bots, and the IoT are new domains relatively to Mobile Apps and Security. Therefore, we see more discussions around these latter domains than the three more recent domains.

With respect to the two difficulty metrics, IoT topics show values similar to big data and chat-bot for one metric (% W/o Accepted Answer). However, among these three recent domains, the IoT have the lowest median time (in hours) to get an accepted answer (2.9 for the IoT, 3.3 for big data, and 14.8 for chat-bot), which shows that IoT developers in SO are relatively more active and engaged than those in the other two domains.

Table 8: Comparison of question types between different domains

Question Type	How	What	Why	Others
IoT	47.3%	37.9%	20%	8.3%
Chatbot	61.8%	11.7%	25.4%	1.2%

Overall, the difference between IoT and chat-bot in terms of the median hours to get an accepted answer is 11.9. This difference is major because, among the five compared domains, big data has the second longest median time of 3.3 hours. Such a difference between IoT and chat bot could be due to the types of OP questions, which are summarized in Table 8. While only 47.3% of questions are of type How in IoT, they are 61.8% in chat bot. In addition, the distribution of questions of type What in IoT is 37.9% and only 11.7% in chat bot, which means that a large

number of IoT questions in SO are about understanding the IoT paradigm via exploratory questions but not in chat bot. We conclude that IoT developers are more engaged than chat-bot developers to offer answers to questions as well as to share their opinions on the diverse IoT architectures, techniques, and tools.

6 Implications of Findings

Our findings can guide the following four IoT stakeholders: (1) **Builders** to prioritize the development of certain IoT architectures, techniques, and tools, (2) **Developers** to prioritize their learning of IoT techniques, (3) **Educators** to guide the mentoring of IoT topics, (4) **IoT Researchers** to determine the most pressing needs in IoT research, and (5) **IoT Enthusiasts and General Readers** to stay aware of the emerging trends in IoT software ecosystems.

Given the empirical nature of this paper, we infer the implications and recommendations based on what we observed in the developers' discussions. Further validations of the implications could benefit from developers' surveys. However, the diversity of IoT topics as we observed in our study makes it a non-trivial task to design a proper survey and to identify a representative sample of IoT developers. Given that our analysis covers a large volume of IoT posts (53K) with discussions from thousands of IoT developers, the findings can later be used to design and conduct multiple IoT-based surveys by focusing on specific IoT topics and categories. In the following, we discuss our findings by providing references to specific IoT questions and by corroborating our results with literature and current IoT ecosystems.

IoT Builders. Figure 9 shows the popularity and difficulty of IoT topics based on two metrics: 1. difficulty with % W/o Accepted Answer 2. popularity with Average Views. The size of each bubble represents the total number of questions.

The size of the topic Secure Messaging in the Network category is the largest because it represents the greatest number of questions on one topic among all topics. It is also among the most difficult topics, with 66% of its questions without accepted answers. We explain this observation as follows: the challenges to ensure security and protect privacy in the IoT is an active research area [36,46,122]. The limited computing resources and the needs to connect to other devices make the IoT intrinsically vulnerable [35]. Zhang et al. [122] noted that data shared among devices may contain large amount of private and sensitive information.

Our analysis of the questions in the topic Secure Messaging shows that developers face difficulty to enforce security protocols. For example, the OP question $Q_{54411947}$ is : *"I am trying to send a testing value to AWS IoT Shadows but when I upload it to my device it keep saying "Cant Setup SSL Connection Trying to send Data""*. This question was posted one year ago, has more than 1,000 views but no accepted answers. Another question, $Q_{17256199}$, reports as *"Why cannot two XBee units communicate?"*. Consequently, builders could propose usable while secure IoT architectures, techniques, or tools.

IoT Developers. The number of smart devices was 5 billions in 2013 and is projected to be 50 billions by 2020 [27]. These new devices come with increasing capabilities, which require new architectures, techniques, and tools. Thus, developers must stay "up to date" on new, emerging IoT topics. Unfortunately, research

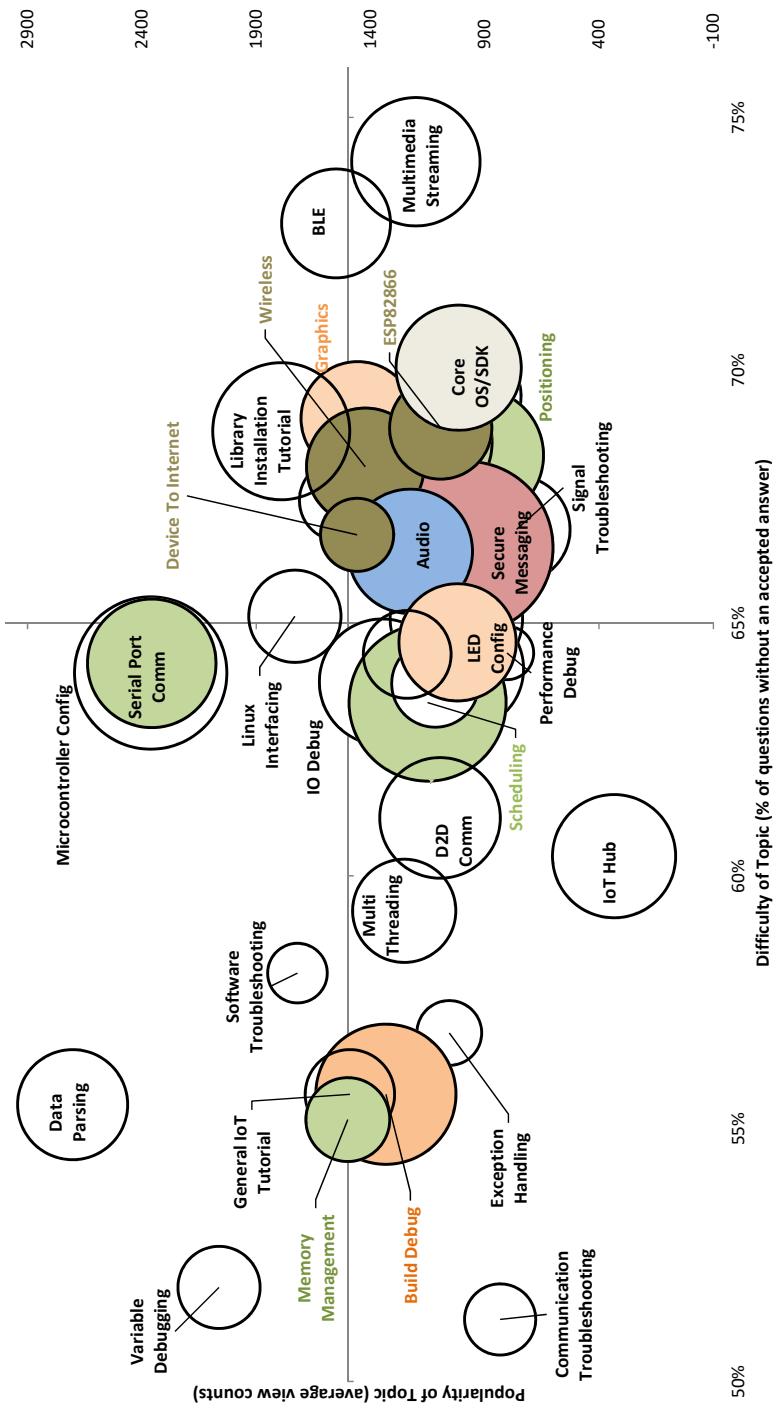

Fig. 9: Tradeoff between IoT topic popularity and difficulty (congested/overlapping topics are colored to distinguish)

showed that orientation in a new domain is difficult [31]. The trade-off between popularity and difficulty of IoT topics shown in Figure 9 offers guidance to developers who would like to seize the IoT paradigm.

For example, the topic Data Parsing in the Software category is most popular yet one of the least difficult topics. Therefore, a developer could begin her journey by learning about Data Parsing. Given that IoT devices have low memory and low energy resources, developers could then focus on building and deploying applications by learning the Memory Management topic in the Hardware category. The developer then could enable the communication among devices by learning from the Communication Troubleshooting topic in the Network category, which is also quite popular yet not difficult. Finally, developers could learn the principles of parallelism in IoT devices based on the topic Multi-threading in Software.

IoT Educators. One of the most popular and least difficult topics in Figure 9 is General IoT Tutorial, which contains questions about IoT basics, such as communication between an Arduino device and a C# program ($Q_{26929153}$) or storing record in Microsoft SQL Server ($Q_{52725652}$). Many of these tutorial questions have more than 1,000 views, showing their popularity. For example, $Q_{14546947}$ OP reports: *"I am new to programming ATtiny chips. I ran the equivalent program to this on an Arduino and it worked, but when running it on an ATtiny2313, although no error message appears, the program appears to freeze."* Such questions show the need for tutorials for (new) IoT developers.

Figure 5 shows the distribution of the four high-level categories based on their numbers of new questions per year. While the arrival of new questions is decreasing for Hardware and Network in 2018, the number of Tutorials questions has been increasing. One reason for the decrease could be that the last major release of the Raspberry PI (3B) happened in 2016.

A close observation also shows that a majority of new questions for both Network and Hardware during this period remain unanswered or without accepted answers. For example, question $Q_{48023866}$, about Wireless Networking, was questioned by OP (Original Poster) two years ago: *"Activating an additional USB WiFi Adapter"*. The original poster acknowledges that she is not Linux savvy: *"I'm trying to add a wifi hotspot/access point to my raspberry pi running Android Things OS... Unfortunately, I am not linux savvy ..."*. It remains without an answer till today. As the IoT paradigm evolves and matures, the needs for tutorials will continue to increase, even more so because official tutorials are often incomplete [104].

Therefore, IoT educators could produce more tutorials to assist (new) IoT developers, in particular based on the difficulty of the topics. For example, Figure 9 shows that tutorials would be particularly useful for Library Installation (in the Tutorials category). This topic has 3.7% of all questions but 67% of those questions remain without accepted answers. An example of such OP question $Q_{28723834}$ is: *"can't install npm onoff on raspberrypi?"*.

IoT Researchers. The popularity of the topic Data Parsing in the Software category could encourage the ongoing research in data analytics in the IoT [56, 106]. More research is needed to lessen the difficulty of some topics. For example, the middle part of Figure 9 shows topics that are difficult yet popular, such as D2I, ESP82866, GPS Positioning, Sensor Feeds, and Secure Messaging. Most of these topics belong to the Network and Hardware categories, which means that these two

categories seem to be, generally, more difficult than the two categories Software and Tutorials.

Figure 5 shows that developers ask increasingly more across all four IoT topic categories. As we observed in Section 4.3, questions of both types How and What are prevalent across all categories but one (i.e., Tutorials), which implies that developers are discussing challenges related to their IoT development in SO. Regarding the Network category, one major difficulty seems to be the lack of proper support and adoption of Secure Messaging protocols. For Hardware, it seems mainly the lack of support for particular MCU (e.g., ESP8266). Therefore, research in IoT could ensure that the technologies are well-supported for the categories. In addition, IoT research could take cues from crowd-source and big-data research to develop techniques that can automatically find acceptable answers to unanswered questions, e.g., by recommending acceptable answers to a question [9, 115].

IoT Enthusiasts can be practitioners/stakeholders who do not necessarily develop or investigate IoT-based solutions, but nevertheless would like to be aware of the state-of-the-art of IoT technologies. Such IoT enthusiasts encompass a large variety of stakeholders, including policy makers, general readers, etc. Indeed, the rapid emergence of IoT-based solutions in our daily life makes it interesting for general IoT readers to be informed of IoT trends. The 40 IoT topics that we observed can be tracked over time to show how they were discussed and evolved over time. In particular, IoT enthusiasts can take note of the evolution of the four topic categories (Software, Network, Hardware, Tutorial). As we have shown in Figure 5 of Section 4.2, all the four topic categories show a steady growth in SO, with the topics related to Software and Network experiencing the most growth. Such insights can inform IoT enthusiasts that topics related to IoT software and networking are the most discussed by developers. As we discussed in Section 4.1 (Figure 3), the Secure Messaging topic from the Network category has the greatest number of questions. In Figure 9 and in Section 4.4, we further showed that the Secure messaging topic is among the most popular in terms of page views, yet it remains one of most difficult. Such information can be useful to make career choice and to recruit non-IoT domains, e.g., hire more security professionals for IoT tools and techniques.

7 Threats to Validity

We now discuss threats to the validity of our study and its results, following common guidelines for empirical studies [113].

External Validity. Threats to external validity concern the generalizability of our findings. We focused on SO, which is one of the largest and most popular developers' Q&A Web sites. Yet, our findings may not generalize to other Q&A Web sites. We only considered questions and accepted answers in our topic modeling. Our approach is consistent with previous work that used topic modeling on SO data [9, 12, 72, 80, 83, 98, 116, 121]. As we noted in Section 5.2, it is difficult to decide (automatically or manually) whether an unaccepted answer is relevant, if we do not know the opinion of the OP about the answer. Even without the excluded (i.e., unaccepted) answers, our studied dataset is quite large (53K posts = questions + accepted answers) and it covers posts over a time period of almost 10 years (2008 –

2019). Therefore, it is possible that a topic in an excluded post could already have been covered in our studied 53K posts. Nevertheless, we accept the threat/risk that we could have missed relevant unaccepted answers and topics.

Internal Validity. Threats to internal validity concern experimental bias and errors while conducting the analysis. In particular, in our study, we manually labeled the topics. To reduce any bias in this labeling, two different authors separately labeled the topics and then another author, who is a domain expert, validated the labeling. The three authors discussed any conflict and resolved them via discussions. Thus, we believe that we reduced labeling bias to an acceptable minimum.

Construct Validity. Threats to construct validity relate to potential errors that may occur when extracting data about IoT-related discussions. We collected all SO posts labeled with one or more tags related to IoT, i.e., 75 different tags. We created the list of tags using state-of-the-art approaches [12,119] and by manually verifying the tags as discussed in Section 3. The tag expansion algorithm uses three initial tags (iot, arduino, and raspberry-pi).We picked the two tags (arduino and raspberry-pi) by analyzing the top 20 tags that co-occurred with the 'iot' tag in SO. Indeed, arduino and raspberry-pi are two of the most popular IoT platforms/tools.In Section 5.3, we offer details on the three initial tag selection processes by also discussing the relevant tag selection technique in SO.

Threats to costruct validity also pertain to the difference between theory, observation, and results. Our use of metrics to measure popularity and difficulty fall under such threats. Yet, we used metrics that were used in previous works [1, 13], thus mitigating the risk of wrong measurements.

8 Conclusions

In this paper, we analyzed IoT-related discussions on Stack Overflow (SO) and applied topic modeling to determine the discussion topics. We present several findings. First, IoT developers discuss a range of topics in SO related to Software, Network, Hardware, and Tutorials. Second, the topic of Secure Messaging among IoT devices in the Network category is the most prevalent topic (i.e., having the most number of questions), followed by the topic of Script Scheduling in the Software category. Third, all the categories are evolving rapidly, i.e., new questions are added at an increasing pace in SO about IoT architectures, techniques, and tools.

Fourth, questions of type How are asked across the three categories Software, Network, and Hardware, although a large number of questions are also of type What. IoT developers are using SO not only to discuss how to solve IoT-related problems but to learn different IoT-related architectures, techniques, and tools. Fifth, topics related to Micro-controller Configuration, IoT serial port communication, and Data Parsing are the most popular. Sixth, topics related to cloud and OS-based IoT software development (e.g., IoT Hub, Windows IoT, and Linux Interfacing) are the most difficult, followed by Hardware-related topic (ESP8266 configuration) and the use of Bluetooth Low Energy (BLE) devices.

Our study opens the doors for the different IoT stakeholders to improve IoT architectures, techniques, and tools. IoT builders can use these findings to provide better support and documentation, developers and educators can use our findings for planning curricula and training, and researchers can direct their focus on the

difficult topics. Tools can be built to support the continuous monitoring and evolution of the IoT topics to make IoT enthusiasts and general reader aware of this rapidly emerging technological landscape.

In the future, we plan to extend our study to focus on individual topics and perform studies focusing on the most difficult topics, e.g., conducting surveys of IoT developers to understand and gain deeper insights about the 40 topics we observed in our empirical study.

Acknowledgment

We sincerely thank the anonymous EMSE reviewers, who helped to significantly improve our paper in the revised manuscript with comments and suggestions.

References

1. A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, and E. Shihab. Challenges in chatbot development: A study of stack overflow posts. In *17th International Conference on Mining Software Repositories, October 5–6, 2020, Seoul, Republic of Korea. New York, NY, USA*. ACM, 2020.
2. A. Agrawal, W. Fu, and T. Menzies. What is wrong with topic modeling? and how to fix it using search-based software engineering. *Information and Software Technology*, 98:74–88, 2018.
3. S. Ahmed and M. Bagherzadeh. What do concurrency developers ask about?: A large-scale study using stack overflow. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, page Article No. 30, 2018.
4. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 2015.
5. M. Aly, F. Khomh, and S. Yacout. What do practitioners discuss about iot and industry 4.0 related technologies? characterization and identification of iot and industry 4.0 categories in stack overflow discussions. *Internet of Things*, 14:100364, 2021.
6. Andrew Kachites McCallum. *MALLET: A Machine Learning for Language Toolkit*. http://mallet.cs.umass.edu/, 2019.
7. D. Andrzejewski, A. Mulhern, B. Liblit, and X. Zhu. Statistical debugging using latent topic models. In *European conference on machine learning*, pages 6–17. Springer, 2007.
8. R. Arun, V. Suresh, C. E. V. Madhavan, and M. N. N. Murthy. On finding the natural number of topics with latent dirichlet allocation: some observations. In *Proceedings of the 14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, pages 391–402, 2010.
9. M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, and K. A. Schneider. Answering questions about unanswered questions of stack overflow. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, pages 87–100, 2013.
10. H. U. Asuncion, A. U. Asuncion, and R. N. Tylor. Software traceability with topic modeling. In *Proc. 32nd Intl. Conf. Software Engineering*, pages 95–104, 2010.
11. L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.
12. M. Bagherzadeh and R. Khatchadourian. Going big: A large-scale study on what big data developers ask. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2019, pages 432–442, New York, NY, USA, 2019. ACM.
13. M. Bagherzadeh and R. Khatchadourian. Going big: a large-scale study on what big data developers ask. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 432–442, 2019.

14. A. Bandeira, C. A. Medeiros, M. Paixao, and P. H. Maia. We need to talk about microservices: an analysis from the discussions on stackoverflow. In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, pages 255–259. IEEE, 2019.
15. A. Barua, S. W. Thomas, and A. E. Hassan. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, pages 1–31, 2012.
16. G. Bavota, M. Gethers, R. Oliveto, D. Poshyvanyk, and A. d. Lucia. Improving software modularization via automated analysis of latent topics and dependencies. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 23(1):1–33, 2014.
17. G. Bavota, R. Oliveto, M. Gethers, D. Poshyvanyk, and A. D. Lucia. Methodbook: Recommending move method refactorings via relational topic models. *IEEE Transactions on Software Engineering*, 40(7):671–694, 2014.
18. L. R. Biggers, C. Bocovich, R. Capshaw, B. P. Eddy, L. H. Etzkorn, and N. A. Kraft. Configuring latent dirichlet allocation based feature location. *Journal Empirical Software Engineering*, 19(3):465–500, 2014.
19. D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.
20. D. M. Blei and J. D. Lafferty. A correlated topic model of science. *The Annals of Applied Science*, 1(1):17–35, 2007.
21. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003.
22. T. Booth, S. Stumpf, J. Bird, and S. Jones. Crossed wires: Investigating the problems of end-user developers in a physical computing task. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3485–3497, 2016.
23. C. Bridge. Unstructured data and the 80 percent rule. *Tersedia di: http://www. clarabridge. com/default. aspx*, 2011.
24. A. Bukhari and X. Liu. A web service search engine for large-scale web service discovery based on the probabilistic topic modeling and clustering. *Service Oriented Computing and Applications*, 12(2):169–182, 2018.
25. B. K. Chae. The evolution of the internet of things (iot): A computational text analysis. *Telecommunications Policy*, 43(10):101848, 2019.
26. M. A. Chaqfeh and N. Mohamed. Challenges in middleware solutions for the internet of things. In *International Conference on Collaboration Technologies and Systems (CTS)*, pages 21–26, 2012.
27. J. Chase. The evolution of internet of things. Technical report, Texas Instruments, 2013.
28. T.-H. Chen, S. W. Thomas, M. Nagappan, and A. E. Hassan. Explaining software defects using topic models. In *9th working conference on mining software repositories*, pages 189–198, 2012.
29. T.-H. P. Chen, S. W. Thomas, and A. E. Hassan. A survey on the use of topic models when mining software repositories. *Empirical Software Engineering*, 21(5):1843–1919, 2016.
30. B. Cleary, C. Exton, J. Buckley, and M. English. An empirical analysis of information retrieval based concept location techniques in software comprehension. *Empirical Software Engineering*, 14:93–130, 2009.
31. B. Dagenais, H. Ossher, R. K. E. Bellamy, and M. P. R. amd Jacqueline P. de Vries. Moving into a new software project landscape. In *32nd ACM/IEEE International Conference on Software Engineering*, pages 275–284, 2010.
32. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.
33. B. Dit, M. Revelle, M. Gethers, and D. Poshyvanyk. Feature location in source code: a taxonomy and survey. *Journal of software: Evolution and Process*, 25(1):53–95, 2013.
34. B. Dit, M. Revelle, and D. Poshyvanyk. Integrating information retrieval, execution and link analysis algorithms to improve feature location in software. *Empirical Software Engineering*, 18(2):277–309, 2013.
35. D. Fahland, D. Lo, and S. Maoz. Mining branching-time scenarios. In *Proc. IEEE/ACM international conference on Automated software engineering*, pages 443–453, 2013.
36. M. Frustaci, P. Pace, G. Aloi, and G. Fortino. Evaluating critical security issues of the iot world: Present and future challenges. *IEEE Internet of Things Journal*, 5(4):2483 – 2495, 2017.

37. Y. Girdhar, P. Giguere, and G. Dudek. Autonomous adaptive underwater exploration using online topic modeling. In *Experimental Robotics*, pages 789–802. Springer, 2013.
38. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.
39. J. Han, E. Shihab, Z. Wan, S. Deng, and X. Xia. What do programmers discuss about deep learning frameworks. *EMPIRICAL SOFTWARE ENGINEERING*, 2020.
40. L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88, 2010.
41. J. Hu, X. Sun, D. Lo, and B. Li. Modeling the evolution of development topics using dynamic topic models. In *IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering*, pages 3–12, 2015.
42. W. Hudson. Card sorting. In M. Soegaard and R. F. Dam, editors, *The Encyclopedia of Human-Computer Interaction*. The Interaction Design Foundation, 2 edition, 2013.
43. A. Kamilaris and N. Botteghi. The penetration of internet of things in robotics: Towards a web of robotic things. *arXiv preprint arXiv:2001.05514*, 2020.
44. K. Kang, J. Choo, and Y. Kim. Whose opinion matters? analyzing relationships between bitcoin prices and user groups in online community. *Social Science Computer Review*, 38(6):686–702, 2020.
45. M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1):81–93, 1938.
46. M. A. Khan and K. Salah. Iot security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82:395–411, 2018.
47. W. H. Kruskal. Historical notes on the wilcoxon unpaired two-sample test. *Journal of the American Statistical Association*, 52:356–360, 1957.
48. S.-E. Lee, M. Choi, and S. Kim. How and what to study about iot: Research trends and future directions from the perspective of social science. *Telecommunications Policy*, 41(10):1056–1067, 2017.
49. H. Li, T.-H. P. Chen, W. Shang, and A. E. Hassan. Studying software logging using topic models. *Empirical Software Engineering*, 23:2655–2694, 2018.
50. Y. Liao, E. de Freitas Rocha Loures, and F. Deschamps. Industrial internet of things: A systematic literature review and insights. *IEEE Internet of Things Journal*, 5(6):4515–4525, 2018.
51. E. Linstead, S. Bajracharya, T. Ngo, P. Rigor, C. Lopes, and P. Baldi. Sourcerer: Mining and searching internet-scale software repos. *Data Min. Knowl Disc.*, 18(2):300–326, 2009.
52. M. Linton, E. G. S. Teo, E. Bommes, C. Chen, and W. K. Härdle. Dynamic topic modelling for cryptocurrency community forums. In *Applied Quantitative Finance*, pages 355–372. Springer, 2017.
53. B. Liu. Sentiment analysis and subjectivity. In N. Indurkhya and F. J. Damerau, editors, *Handbook of Natural Language Processing*. CRC Press, Taylor and Francis Group, 2nd edition, 2016.
54. L. Liu, L. Tang, W. Dong, S. Yao, and W. Zhou. An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus*, 5(1):1608, 2016.
55. X. Liu, X. Sun, B. Li, and J. Zhu. Pfn: A novel program feature network for program comprehension. In *2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS)*, pages 349–354. IEEE, 2014.
56. M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob. Big iot data analytics: Architecture, opportunities, and open research challenges. *IEEE Access*, 5(1):5247 – 5261, 2017.
57. E. Mathews, S. S. Guclu, Q. Liu, T. Ozcelebi, and J. J. Lukkien. The internet of lights: An open reference architecture and implementation for intelligent solid state lighting systems. *Energies*, 10(8):1187, 2017.
58. M. L. McHugh. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282, 2012.
59. T. Mens, A. Serebrenik, and A. Cleve. *Evolving Software Systems*, volume 190. Springer, 2014.
60. A. U. Mentsiev, A. U. Mentsiev, and E. F. Amirova. Iot and mechanization in agriculture: problems, solutions, and prospects. *IOP Conference Series: Earth and Environmental Science*, 548(3):032035, 2020.
61. D. Minoli, K. Sohraby, and B. Occhiogrosso. Iot security (IoTSec) mechanisms for e-health and ambient assisted living applications. In *IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies*, pages 13–18, 2017.

62. D. Mocrii, Y. Chen, and P. Musilek. Iot-based smart homes: A review of system architecture, software, communications, privacy and security. *Internet of Things*, 1:81–98, 2018.
63. H. Nabli, R. B. Djemaa, and I. A. B. Amor. Efficient cloud service discovery approach based on lda topic modeling. *Journal of Systems and Software*, 146:233–248, 2018.
64. T. T. Nguyen, T. N. Nguyen, and T. M. Phuong. Topic-based defect prediction (nier track). In *Proceedings of the 33rd international conference on software engineering*, pages 932–935, 2011.
65. K. Nie and L. Zhang. Software feature location based on topic models. In *2012 19th Asia-Pacific Software Engineering Conference*, volume 1, pages 547–552. IEEE, 2012.
66. NLTK. *Sentiment Analysis*. http://www.nltk.org/howto/sentiment.html, 2016.
67. S. Overflow. *Stack Overflow Questions*. https://stackoverflow.com/questions/, 2020. Last accessed on 14 November 2020.
68. A. Panichella, B. Dit, R. Oliveto, M. D. Penta, D. Poshyvanyk, and A. D. Lucia. How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms. In *International Conference on Software Engineering*, pages 522–531, 2013.
69. A. Panichella, B. Dit, R. Oliveto, M. D. Penta, D. Poshyvanyk, and A. D. Lucia. Parameterizing and assembling ir-based solutions for se tasks using genetic algorithms. In *23rd IEEE international conference on software analysis, evolution, and reengineering*, 2016.
70. A. R. Pathak, M. Pandey, and S. Rautaray. Adaptive model for dynamic and temporal topic modeling from big data using deep learning architecture. *International Journal of Intelligent Systems and Applications*, 11(6):13, 2019.
71. L. Ponzanelli, G. Bavota, M. Di Penta, R. Oliveto, and M. Lanza. Prompter: Turning the IDE into a self-confident programming assistant. *Empirical Software Engineering*, 21(5):2190–2231, 2016.
72. L. Ponzanelli, A. Mocci, A. Bacchelli, and M. Lanza. Improving low quality stack overflow post detection. In *In Proceedings of the 30th International Conference on Software Maintenance and Evolution*, pages 541–544, 2014.
73. M. F. Porter. An algorithm for suffix stripping. In K. S. Jones and P. K. Willett, editors, *Readings in information retrieval*. Morgan Kaufmann Publishers Inc., 1st edition, 1997.
74. D. Poshyvanyk, M. Gethers, and A. Marcus. Concept location using formal concept analysis and information retrieval. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 21(4):1–34, 2013.
75. D. Poshyvanyk, Y.-G. Guéhéneuc, A. Marcus, G. Antoniol, and V. T. Rajlich. Feature location using probabilistic ranking of methods based on execution scenarios and information retrieval. *IEEE Transactions on Software Engineering*, 33(6):420–432, 2007.
76. K. Pretz. The next evolution of the internet. *IEEE Magazine The institute*, 50(5), 2013.
77. L. F. Rahman, T. Ozcelebi, and J. Lukkien. Understanding iot systems: a life cycle approach. *Procedia computer science*, 130:1057–1062, 2018.
78. S. Rao and A. C. Kak. Retrieval from software libraries for bug localization: a comparative study of generic and composite text models. In *8th Working Conference on Mining Software Repositories*, page 43–52, 2011.
79. R. Řehůřek and P. Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.
80. X. Ren, Z. Xing, X. Xia, G. Li, and J. Sun. Discovering, explaining and summarizing controversial discussions in community q&a sites. In *34th IEEE/ACM International Conference on Automated Software Engineering*, pages 151–162, 2019.
81. M. Röder, A. Both, and A. Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 399–408, 2015.
82. C. Rosen and E. Shihab. What are mobile developers asking about? a large scale study using stack overflow. *Empirical Software Engineering*, page 33, 2015.
83. C. Rosen and E. Shihab. What are mobile developers asking about? a large scale study using stack overflow. *Journal Empirical Software Engineering*, 21(3):1192–1223, 2016.
84. C. Rosen and E. Shihab. What are mobile developers asking about? a large scale study using stack overflow. *Empirical Software Engineering*, 21(3):1192–1223, 2016.
85. G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of American Society for Information Science*, 41(4):288–297, 1990.
86. T. Savage, B. Dit, M. Gethers, and D. Poshyvanyk. Topic xp: Exploring topics in source code using latent dirichlet allocation. In *2010 IEEE International Conference on Software Maintenance*, pages 1–6. IEEE, 2010.

87. P. Sethi and S. R. Sarangi. Internet of things: Architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, 2017, 2017.
88. M. N. Shahid. A cross-disciplinary review of blockchain research trends and methodologies: topic modeling approach. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020.
89. N. Sharma, M. Shamkuwar, and I. Singh. The history, present and future with iot. *Internet of Things and Big Data Analytics for Smart Generation*, 154(1):27–51, 2019.
90. S. Singh, P. K. Sharma, B. Yoon, M. Shojafar, G. H. Cho, and I.-H. Ra. Convergence of blockchain and artificial intelligence in iot network for the sustainable smart city. *Sustainable Cities and Society*, 63:102364, 2020.
91. Stack Exchange, Inc. *Stack Exchange Data Dump*. https://archive.org/details/stackexchange, 2019.
92. Stack Overflow. *Statistics: What is the average response time on Stack Overflow?* https://meta.stackexchange.com/questions/61301, 2010.
93. Stack Overflow. *Tags*. https://stackoverflow.com/tags, 2021.
94. M. Steyver and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.
95. X. Sun, B. Li, H. Leung, B. Li, and Y. Li. Msr4sm: Using topic models to effectively mining software repositories for software maintenance tasks. *Information and Software Technology*, 66:671–694, 2015.
96. X. Sun, B. Li, Y. Li, and Y. Chen. What information in software historical repositories do we need to support software maintenance tasks? an approach based on topic model. *Computer and Information Science*, pages 22–37, 2015.
97. X. Sun, X. Liu, B. Li, Y. Duan, H. Yang, and J. Hu. Exploring topic models in software engineering data analysis: A survey. In *17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 357–362, 2016.
98. V. Terragni, Y. Liu, and S.-C. Cheung. Csnippex: automated synthesis of compilable code snippets from q&a sites. In *In Proceedings of the 25th International Symposium on Software Testing and Analysis*, pages 118–129, 2016.
99. S. W. Thomas, B. Adams, A. E. Hassan, and D. Blostein. Modeling the evolution of topics in source code histories. In *8th working conference on mining software repositories*, pages 173–182, 2011.
100. S. W. Thomas, B. Adams, A. E. Hassan, and D. Blostein. Studying software evolution using topic models. *Science of Computer Programming*, 80(B):457–479, 2014.
101. K. Tian, M. Revelle, and D. Poshyvanyk. Using latent dirichlet allocation for automatic categorization of software. In *6th international working conference on mining software repositories*, pages 163–166, 2009.
102. G. Uddin, O. Baysal, L. Guerrouj, and F. Khomh. Understanding how and why developers seek and analyze API-related opinions. *IEEE Transactions on Software Engineering*, page 37, 2018. Under review.
103. G. Uddin and F. Khomh. Automatic summarization of API reviews. In *Proc. 32nd IEEE/ACM International Conference on Automated Software Engineering*, page 12, 2017.
104. G. Uddin and M. P. Robillard. How api documentation fails. *IEEE Softawre*, 32(4):76–83, 2015.
105. I. Vayansky and S. A. Kumar. A review of topic modeling methods. *Information Systems*, 94:101582, 2020.
106. S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato. A survey on network methodologies for real-time analytics of massive iot data and open research issues. *IEEE Communications Surveys & Tutorials*, 19(3):1457 – 1477, 2017.
107. Z. Wan, X. Xia, and A. E. Hassan. What do programmers discuss about blockchain? a case study on the use of balanced lda and the reference architecture of a domain to capture online discussions about blockchain platforms across stack exchange communities. *IEEE Transactions on Software Engineering*, 1(1):24, 2019.
108. Z. Wan, X. Xia, and A. E. Hassan. What is discussed about blockchain? a case study on the use of balanced lda and the reference architecture of a domain to capture online discussions about blockchain platforms across the stack exchange communities. *IEEE Transactions on Software Engineering*, 2019.
109. J. Wang, P. Gao, Y. Ma, K. He, and P. C. Hung. A web service discovery approach based on common topic groups extraction. *IEEE Access*, 5:10193–10208, 2017.

110. S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang. Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, 101:158–168, 2016.
111. M. Weyrich and C. Ebert. Reference architectures for the internet of things. *IEEE Software*, 33(1):112–116, 2016.
112. A. Whitmore, A. Agarwal, and L. Da Xu. The internet of things—a survey of topics and trends. *Information systems frontiers*, 17(2):261–274, 2015.
113. C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering: an introduction*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.
114. X. Xie, W. Zhang, Y. Yang, and Q. Wang. Dretom: Developer recommendation based on topic models for bug resolution. In *Proceedings of the 8th international conference on predictive models in software engineering*, pages 19–28, 2012.
115. B. Xu, Z. Xing, X. Xia, and D. Lo. Answerbot: automated generation of answer summary to developers' technical questions. In *Proc. 32nd IEEE/ACM International Conference on Automated Software Engineering*, pages 706–716, 2017.
116. D. Yang, A. Hussain, and C. V. Lopes. From query to usable code: an analysis of stack overflow code snippets. In *In Proceedings of the 13th International Conference on Mining Software Repositories*, pages 391–402, 2016.
117. G. Yang, T. Zhang, and B. Lee. Towards semi-automatic bug triage and severity prediction based on topic model and multi-feature of bug reports. In *2014 IEEE 38th Annual Computer Software and Applications Conference*, pages 97–106. IEEE, 2014.
118. X.-L. Yang, D. Lo, X. Xia, Z.-Y. Wan, and J.-L. Sun. What security questions do developers ask? a large-scale study of stack overflow posts. *Journal of Computer Science and Technology*, 31(5):910–924, 2016.
119. X.-L. Yang, D. Lo, X. Xia, Z.-Y. Wan, and J.-L. Sun. What security questions do developers ask? a large-scale study of stack overflow posts. *Journal of Computer Science and Technology*, 31(5):910–924, 2016.
120. Z. Yang, Y. Yue, Y. Yang, Y. Peng, X. Wang, and W. Liu. Study and application on the architecture and key technologies for IoT. In *International Conference on Multimedia Technology*, pages 747–751, 2011.
121. T. Zhang, G. Upadhyaya, A. Reinhardt, H. Rajan, and M. Kim. Are code examples on an online q&a forum reliable?: a study of api misuse on stack overflow. In *In Proceedings of the 40th International Conference on Software Engineering*, pages 886–896, 2018.
122. Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh. Iot security: Ongoing challenges and research opportunities. In *IEEE 7th International Conference on Service-Oriented Computing and Applications*, pages 230–234, 2014.
123. Y. Zheng, Y.-J. Zhang, and H. Larochelle. A deep and autoregressive approach for topic modeling of multimodal data. *IEEE transactions on pattern analysis and machine intelligence*, 38(6):1056–1069, 2015.