ThingSpeak in the Wild: Exploring 38K Visualizations of IoT Data

Thomas Zachariah
University of California, Berkeley
tzachari@berkeley.edu

Noah Klugman
nLine, Inc.
noah@nline.io

Prabal Dutta
University of California, Berkeley
prabal@berkeley.edu

ABSTRACT

Cloud services are vital tools for storing, analyzing, and responding to data collected by sensors in the Internet of Things (IoT). With ThingSpeak, a popular web platform that provides these services, users can easily create cloud "channels" to receive, host, and visualize sensor data. In this study, we scrape public channels from 6,511 users to construct a comprehensive picture of both the ThingSpeak developer community and their applications. We release this data to support future work. From this data, we examine 37,989 visualizations and uncover relationships between application domains and visualization techniques utilized. Further, we investigate how ThingSpeak's interface impacts user design choices. To learn which channels most successfully disseminate information, we explore design patterns on channels "liked" on Facebook or discussed in ThingSpeak forums. Finally, we briefly comment on how services like ThingSpeak can better support users' needs moving forward.

CCS CONCEPTS

• **Human-centered computing → Visualization**.

KEYWORDS

Internet of Things, Cloud, PaaS, Visualization, Dataset, Analysis

ACM Reference Format:
Thomas Zachariah, Noah Klugman, and Prabal Dutta. 2022. ThingSpeak in the Wild: Exploring 38K Visualizations of IoT Data. In *The 20th ACM Conference on Embedded Networked Sensor Systems (SenSys '22).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3560905.3567766

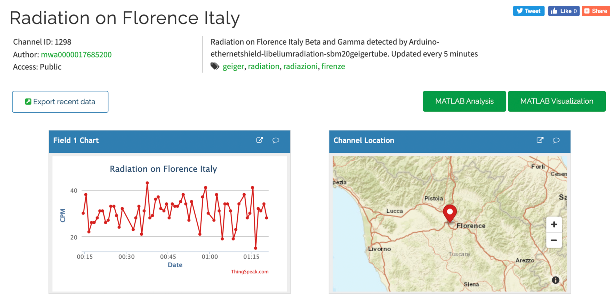

Figure 1: Example public ThingSpeak channel. Data collected include title, description, tags, social media likes, comments, and visualization types.

1 INTRODUCTION

IoT systems largely have a common architecture — *sensors* send data through a *gateway* to the *cloud*. Applications often rely on the cloud for at least five key services — receiving, storing, processing, visualizing, and responding to data collected. These services have been repeatedly bundled and released commercially under a platform-as-a-service (PaaS) model [3, 4, 10, 12, 23, 39]. As a result, many IoT applications are now centralized — developers from all areas of IoT are using the same services. This provides an unprecedented opportunity to study a comprehensive range of application domains, and answer questions about participants in the IoT landscape and types of data collected. In this work, we compile and analyze a significant public dataset scraped from an IoT PaaS cloud. We believe this dataset and analysis can be used as resources to better understand how IoT cloud platforms are constructed and how design choices may lead to meaningful insights from IoT data.

The IoT PaaS we examine, ThingSpeak [39], allows IoT developers, both in the hobby and commercial space, to collect, analyze, and visualize data from sensors [15]. It was created in 2010, acquired by Mathworks in 2014, and had over 60,000 users by 2016 [5]. With ThingSpeak, developers are able to openly publish their data visualizations in "public channels". As such, we find the site provides more accessible insight about the applications it is running than counterparts with larger user-bases like AWS IoT and Google IoT.

We focus our analysis on both the IoT applications and the developers themselves. For both groups we get our data by scraping the ThingSpeak website. Our scraper, the resultant ThingSpeak dataset, and scripts used for analysis are made publicly available [17]. Our analysis of IoT applications explores 9,078 public active ThingSpeak channels created by 6,511 users and containing 37,989 visualizations, descriptions/tags further describing the application, public channel comments, and number of social media likes. Our analysis of developers looks at the type, number, and quality of applications each has created, comments made on channels and developer forums, and descriptions and tags chosen for their content.

From this analysis we highlight some key observations. First, there is little diversity in the types of applications—manual classification of active, English public channels returns only 15 significant categories, bringing into question how the platform's design choices might affect what developers consider possible. Second, by scraping forum posts and Facebook "likes", we reject the hypothesis that popular channels are more novel or better described than less popular channels, instead finding few patterns that explain community interest. Finally, we look at the types of visualizations used for each category of channel activity and do not find category-specific visualizations, indicating that users may not think deeply about how best to visualize their unique data streams, or, just that different application domains have similar data visualization requirements.

These insights, along with others throughout the paper, are enabled for the first time due to the ThingSpeak dataset, which combines low-level information about an IoT application with information about the developer who created the application, ultimately providing new insights into how the IoT functions in the wild.

SenSys '22, November 6–9, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9886-2/22/11.
https://doi.org/10.1145/3560905.3567766

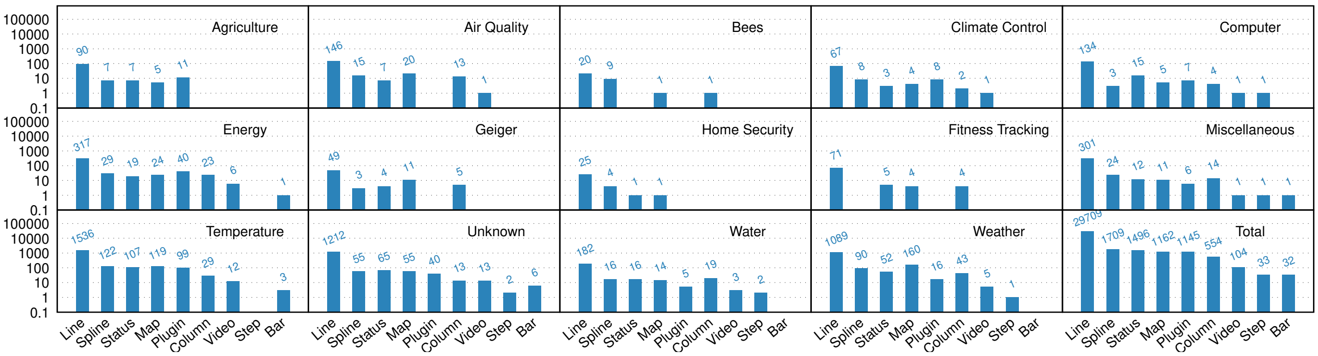

Figure 2: Visualization types by application. By default, ThingSpeak allows users to insert time-series *Line* graphs, deployment *Map*, channel *Video*, and a *Status* indicator. Additionally, Mathworks' Matlab is used to make *Spline*, *Bar*, *Step*, and *Column* plots. A *Plugin* can be added to run arbitrary Javascript with visual output. Listed application categories account only for visualizations on English-language channels. "Total" presents a breakdown for all public channels.

2 RELATED WORK

ThingSpeak is a web service for collection, storage, and visualization of data, particularly targeting inputs from embedded sensor devices [21, 29, 39]. It is utilized for applications like environmental monitoring [9, 26], energy management [2], home automation [27], health tracking [30], and more [1]. It also functions as an educational tool, supporting IoT workshops and early STEM [7, 8].

ThingSpeak is only one of a number of emerging IoT PaaS providers [3, 4, 6, 10, 12, 13, 23, 28]. While many of these provide similar services, only some introduce a community where public applications can be hosted [3, 13]. Thingful attempts to aggregate public IoT data into a "search engine for the IoT". Unfortunately, the data made available from Thingful lacks much of the meta-data and visualization capacities of ThingSpeak or similar services [40].

Work from Maureira et al. in 2011 evaluates ThingSpeak with a focus on the underlying computing services [21]. This includes a deeper dive into how a user should configure ThingSpeak, and includes code samples for different methods of interacting with various services. While their work touches on a few early applications, we present a more thorough exploration of ThingSpeak's use in the wild by scraping the entire set of active public channels.

Our work is partially inspired by Nandi et. al and Yu et. al [41, 42, 44], who examine usage of IFTTT – a cloud service that provides a simple way to configure and host applets that respond to changes in data – by scraping its public applets. Unlike ThingSpeak, "a minority of IFTTT recipes involve physical devices" [42]. Instead, it provides a structured UI that abstracts away the low-level applet design, preventing insight into IoT developers and their applications.

Like other works that evaluate information dissemination capacities of a web service [16, 33], we analyze how well ThingSpeak channels visualize and describe the data they hold. Like studies that have used passive measurements of user behavior within a web service to provide insights about potential design improvements [19, 22], we identify constraints that the platform might be unintentionally placing on its users, and highlight common patterns and popular features. Our efforts to classify both ThingSpeak users and channels share similar motivations to existing work [43]. This broad related work supports our methodology and directs us as we perform our exploration through the dataset of IoT applications and developer behaviors found on ThingSpeak.

3 METHODOLOGY

We describe our experimental methodology including the characteristics and techniques used to extract and analyze data.

3.1 Data Source

ThingSpeak organizes its platform into "channels", each of which can be thought of as an application. Figure 1 is an example. Users can create and manage multiple channels through the website. Channels serve as destinations for users to send sensor data, write code to analyze and visualize the data, and configure event-triggers. Unlike IFTTT formulas, channels cannot be easily duplicated among users.

The site's free user tier limits incoming messages to 8,200 per day, allows updates every 15 seconds at maximum, and times out analytics after 20 seconds of processing. Purchasing a personal, commercial, or academic license removes many of these limits [36]. Channels can be public or private. Private ones are only accessible to the owner and users with permission. Public channels are accessible by anyone and are a primary source of data scraped for this work.

ThingSpeak provides a channel creation UI, where a user configures the channel name, description/tags (optional), and up to 8 data fields (at least one required). The user can then add charts or other visualizations for that data to the channel. The 8-field constraint leads to some notable usage patterns that we explore later. This user-provided data is not sanitized by ThingSpeak, leaving the user free to provide as much or as accurate information as they desire.

3.2 Data Collection

We developed a software tool to scrape the public channels and forums of the ThingSpeak website. To minimize any impact of the scraping activity, we designed these scripts to visit each subdomain as infrequently as possible. We used the BeautifulSoup library to record notable features like channel description, tags, charts, timestamps, and deployment location [31] from "active" channels — channels that have had recent data enter them — which make up 6.18 % of total public channels. This decision was made to minimize the total data scraped and because active channels represent the best picture of applications that hold sustained value. We collected supplementary information for both active and inactive channels from associated JSON feed files containing metadata.

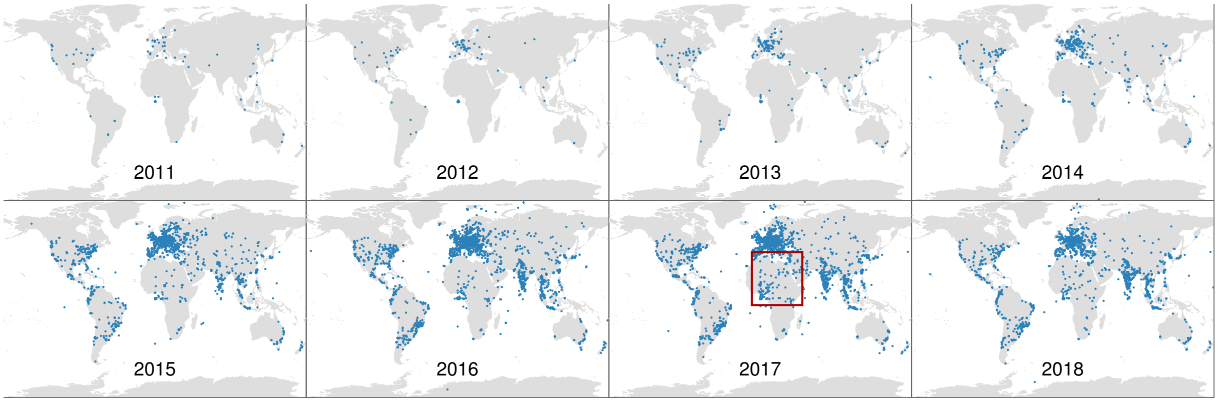

Figure 3: Declared deployment locations over time. Each map displays the user-declared coordinates for each channel. While most points seem to be plausible deployment locations, we note an artifact off the coast of Africa at coordinate <0,0> bearing resemblance to a plot with a 45-degree trend line (most strongly in 2017 and highlighted with a red box). Users frequently reporting <0,0>, <x,0>, or <0,y> indicates inaccurate self-reported location data.

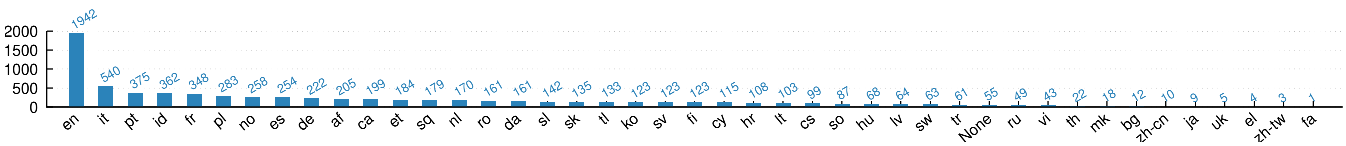

Figure 4: Number of users per language. A diverse set of languages are used — English is the most common at ~30%. Language identification is performed with the Python library langdetect [18] using ISO-639 labels [14]. "None" is the result provided when langdetect is unable to classify the given word sample.

3.3 Data Analysis

We perform our analysis in two dimensions, firstly by exploring across different applications domains emerging on the ThingSpeak platform, and secondly by turning towards the user population to extract common design patterns and to explore the user experience.

To process most of the data, we developed a series of Javascript and Python scripts, each of which are made publicly available. To perform our semantic text analysis and keyword extraction, we used the Microsoft Azure Text Analytics Cognitive Service [24, 25]. This service allows for training free classification of text, and the full extent of our analysis fell within the free tier of the service. To classify the language of the text we used the Python library langdetect [18]. To classify application types we manually labeled all public active channels that contained English language descriptions.

3.4 Data Characteristics

We recognize limitations with our collection of data. First, there may be a selection bias introduced by only considering users who create public channels and public applications. Because we have no access to private channels, we can not quantify the impact of this potential bias. Second, 3.2% of visualizations were not explicitly categorized, as they were broken or consisted of a custom plugin that broke patterns expected by the scraper. These charts have been excluded from the analysis. Third, for semantic analysis, sentiment analysis, and application classification, we exclude all non-English-language channels, with 1942 channels remaining. This is to take advantage of the previously existing semantic and sentiment analysis tools trained on English and because the manual classification of application type from channel descriptions was performed by English speakers. Finally, the code doing data analysis was not scraped, preventing the examination of methods used for data processing.

4 ANALYSIS

We evaluate users and public applications of ThingSpeak and uncover common usage patterns, behaviors, and experiences. Among other findings, we learn that the number of visualizations per channel seems to be more a function of platform constraints than application needs, that every application domain favors line charts, that the words used in tags and descriptions are similar, and that user sentiment in forums tend to be neutral to positive for all topics.

4.1 Applications

We look at how ThingSpeak active public channels are used – identifying a small number of popular application domains – and discover that design patterns for channels in all domains are largely similar.

Labeling We manually grouped all 3900 English active public channels into application types by reading their channel descriptions when present. An earlier attempt to automate this with a semantic unlabeled text clustering system failed due at least in part to a small training set size. Further, the domain specific language in the descriptions reduced the accuracy of pre-trained models [32].

We find that the channels can be grouped into a fairly small (14) set of popular applications (those with cluster size greater than 0.5%), with the top 5 applications (Temperature, Weather, Energy, Water, and Air Quality) making up 61.7% of channels. These application categories are broken down in Figure 2. This smaller grouping may indicate that ThingSpeak inadvertently limits application types or that the descriptions used for this clustering do not provide specific enough resolution to define clear lower level clusters. 6.1% of channels do not fall into clusters and are labeled Miscellaneous. Further, we find that many applications (22.7%) are unable to be classified because their descriptions contain limited information, such as "arduino based project" or "all in one".

Application-specific channel patterns We explore which visualizations are most often used by different application domains, the result of which is shown in Figure 2, and find that each application domain favors line graphs. This may be explained by line graphs being the default choice of visualization presented by the ThingSpeak UI. Another explanation is that line graphs are introduced in many of the ThingSpeak tutorials, and therefore are the familiar choice when building channels. We also observe that no application domain consists of fewer than four chart types, showing a diversity of visualization requirements across all applications.

Popularity To gauge what receives attention, we explore channels that have been "liked" through the in-channel Facebook button. Only ~2 % of channels have one or more likes. We find little correlation between the type of application and number of Facebook likes a channel receives. 50 % of application types have at least one channel with more than 5 likes, placing it above the 90th percentile of overall channels with likes. For example, only one Water channel is "liked", but is also the most liked English public channel. Likes are also not linearly correlated with the number of visualizations it contains. Channels with 4, 5 and 8 visualizations are "liked" more commonly, while channels with 1, 2, or 6 visualizations are avoided, which could indicate a correspondence with perceived value.

4.2 User Population and Behavior

To evaluate user behavior, we examine active public channels of 6,511 users and determine the location, language, creation time, description, tags, and charts used by each. We also consider users' forum posts, channel comments, and number of channels created.

Are there super users? To understand how users engage with ThingSpeak, we first look at the numbers of active public channels created by each user. We classify a super user as a user who creates a number of channels greater than a single standard deviation (2.87) above the mean (1.37) and find 51 super users (1.5 % of the total active public channel user base). The fact that an average user creates less than 3 channels could indicate that the use case for active public ThingSpeak channels is not complex – users only have a small number of devices or a small number of deployments. Alternatively, because nothing in the ThingSpeak architecture enforces the model of a single application per channel, this could indicate that channels are overloaded to contain multiple different applications. Additionally, API limits constrain how much and how often users can send data to channels, particularly in the free-tier service.

Where are users from? By analyzing channel descriptions, we see the geographic spread of ThingSpeak channels over time by collecting the user-reported locations of all active and non-active public channels. These coordinates are mapped in Figure 3. We can observe that ThingSpeak has been adopted and spread across 76 different countries, especially since 2015. Europe and the United States were early adopters, but were followed quickly afterwards by South America, Asia, and Africa. However, we also observe users self-reporting locations incorrectly, with many reporting location at or around <0,0>, a point in the middle of the ocean. We further discovered usage of 43 languages. The number of channels per language of description is shown in Figure 4. We see that English is most commonly used. This may be because ThingSpeak does not offer any official language translation to their platform.

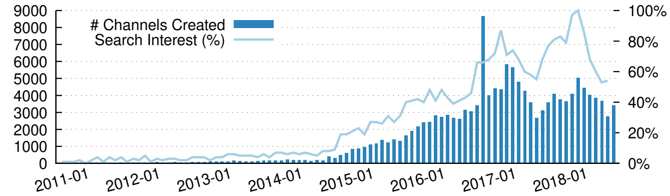

Figure 5: Channel creation & Google search trend [11]. Public interest in Thingspeak and rate of channel creation have grown together over time. The 2016 spike coincides with the Thingspeak Python API release[38], two popular tutorials [34, 35], and a Mathworks blog post on new features [37].

How do users design channels? While ThingSpeak allows for a fair amount of flexibility in channel design, it also implicitly encourages specific design patterns through limitations on user data and user interfaces. For example, the platform influences the visualizations used by providing a well-documented, default line graph visualization. If users are not happy with this line graph they can include other types of line graphs, but the process is difficult technically and not well-documented. As a result, over 82.7 % of total visualizations across active channels use the default line graph.

Another platform-influenced design pattern is revealed through considering the number of charts per channel. By default, the ThingSpeak API allows users to create any number of charts to visualize data, but it only allows users to collect up to 8 data points in a channel per update. With this in mind, we explore how users design visualizations by measuring number of charts used per channel.

Over one-third of channels display either 1 or 2 charts. About 26% use 3 or 4. We observe a dip in the number of channels with 5, 6, or 7 (~7% each). Then, ~11.5% display 8 charts. This suggests a tendency in users to visualize either the minimal or maximum number of data points suggested by the interface. The fact that fewer users choose intermediate points between these options could support a hypothesis that the specific number of charts is not simply a function of the type of data being visualized. Moreover, there is a swift drop-off in the number of channels that include more than 8 charts (~4% total). This signals that users may be limiting their channels to a single chart per data point, rather than using multiple visualizations of the same data point.

Next, we investigate textual annotations tied to each channel, such as descriptions and tags. In ThingSpeak's interface, channel description are optional. 53.1 % of channels have no description, while 3.6 % have a description with five words or fewer. The high percentage of users who chose to not provide channel description may indicate that users do not care about preparing accessible public channels, or that users do not view descriptions as important for communicating information about their channel. Using description length as a rough proxy for description quality, we investigate the distribution of the number of words per descriptions. We find the average word count to be 36.5 with a long tail of 6.2 % of channels using more than 100 words. This distribution shows that users who write descriptions in English most often include enough words to capture meaningful information about the channel.

We construct word clouds to further characterize information contained in channel descriptions and tags. The word clouds for all language descriptions, all language tags, English descriptions, and English keywords – as extracted by Azure from the English

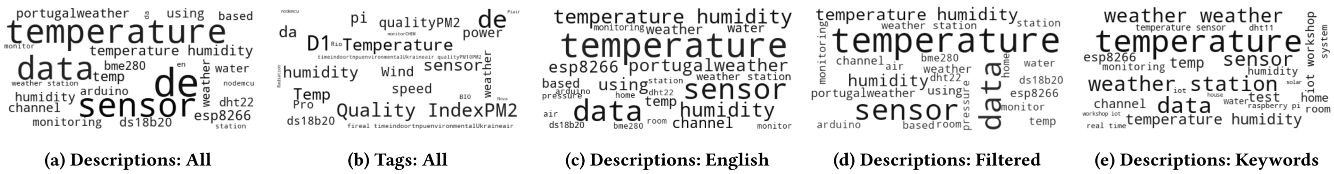

Figure 6: User-inputted tags and descriptions. Each word cloud is constructed with a maximum of 25 words and the same proportional relationship between font size and word frequency. Cloud (d) contains English language descriptions with duplicates of the ThingSpeak tutorial filtered out. Cloud (e) contains keywords auto-extracted using the Microsoft Azure Text Analytics Cognitive Service from the English language channel descriptions [24].

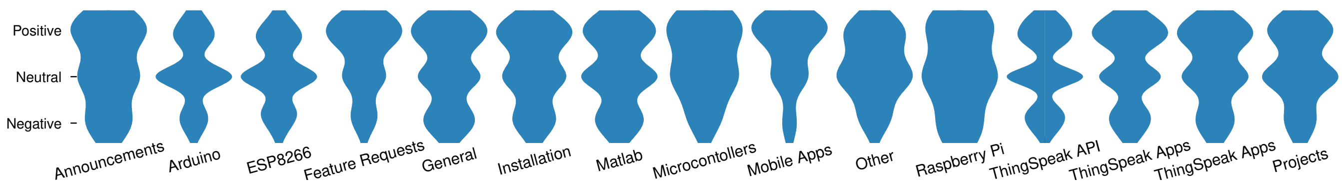

Figure 7: Sentiment of forum posts. Distributions of forum topic sentiment scores, which rank posts from negative to positive. Each forum topic is plotted separately to better understand the context for particular emotions in posts. Sentiment is classified using Azure Text Analytics [24].

descriptions – are shown in Figure 6. Each of these four clouds expose similar keywords, demonstrating that tags, descriptions in different languages, and automatically extracted keywords contain overlapping information. Further, clouds (c) and (d) are more closely related than (a) and (b), showing that keywords extracted from the description can be at least as representative of the application than the tags manually added by a user. This reveals that keyword extraction from channel descriptions is a potential avenue for automated application-type classification.

When were channels built? To better understand ThingSpeak user adoption we explore channel creation trends over time. We first consider the creation across time of all active and not active channels, shown in Figure 5. This graph displays a gradual increase in channel creation over time. A few notable upticks, like the activity near the end of 2016 perhaps suggests that some supplementary influence led to temporary influxes of new channels. To validate this, we gather Google search history for the phrase "ThingSpeak" and find a similarly shaped curve, which is overlaid [11]. We find that the upticks most likely correspond with points of time in which tutorials were launched and the service received media attention.

Who uses the forum? ThingSpeak provides a forum for its developer community, which contains 5703 posts spread from 1283 users among 17 sub-forums. We examine who is using the forum with the assumption that either the users who have created the most channels or the users who have commented on the most channels will be most active in the forum. Instead, we find a very small number of public-channel users have any activity in the forum. Only 8 users both created an active public channel and posted in the forum. Further, only 4 users both posted in the forum and commented on a public channel. This could indicate that public-channel users are less likely to ask for help, or their applications are simpler to develop than other user groups.

Taking a more comprehensive look at participation in the forums, we learn that the top 8 users (0.62% of users) in the forum contributed 34.6 % of all posts, demonstrating that the forum is not broadly used across the developer community, and uncovering a heavy-tail distribution similar to the findings of a study by Lerman et al., in which it is revealed that the top 3% of users on the online forum Digg contribute 33% of the site's submissions [20]. We examine the number of posts per thread and find an average number of 2.93 with a standard deviation of 0.799. This shows that users do not typically use the forum for long discussions, and that no single area of the forum spurs much lengthier discourse than any other.

Are people happy? Although we found above that both public and private channel users are not often involved with the forum, we explore the forum posts to understand the general sentiment of the subset of ThingSpeak users that do engage with the forum, the results of which are shown averaged per user in Figure 7. Sentiment was calculated from the text content of the forum post using Azure Text Analytics [24]. The relatively positive attitude in posts relating to "Feature Requests", "Mobile Apps", and "Microcontrollers" could demonstrate areas where users are most excited or areas where users are most likely to receive helpful support. Additionally, the high favorability in "Plugins", "Matlab", and "Installation" may indicate that users are satisfied with the usability of ThingSpeak.

5 CONCLUSION

ThingSpeak channels have been created by a global population of users and are collecting data on six continents. This, at the highest level, demonstrates the culturally agnostic draw for using technology to gather and react to data. Despite this diversity in user base, channels tend to focus on a small set of popular application domains. This result both serves as a guide to which applications are most popular today and is a striking reminder of how nascent and narrowly applied IoT in the wild still is. From this exploration of user design patterns, we see that the structure of a cloud service may influence how people choose to imagine their IoT. This should be understood as a great responsibility placed on platform providers and is one that should be considered with intention. The restrictions imposed by a platform carry with them a potential risk of promoting frivolous or routine deployments with shallow insights, far from the promise of emerging, deeply-embedded technology that improves quality of life, and as a consequence potentially trivializing what could be the revolutionary digitization of the physical world.

ACKNOWLEDGMENTS

This work was supported in part by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. We thank Meghan Clark, Branden Ghena, Will Huang, Pat Pannuto, Lane Powell, and Jean-Luc Watson for support with this work, and the reviewers for their feedback.

REFERENCES

[1] A. I. Abdul-Rahman and C. A. Graves. 2016. Internet of Things Application Using Tethered MSP430 to Thingspeak Cloud. In *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*. IEEE, New York, NY, USA, 352–357. DOI: https://doi.org/10.1109/SOSE.2016.42

[2] M. Abo-Zahhad, S. M. Ahmed, M. Farrag, M. F. A. Ahmed, and A. Ali. 2015. Design and implementation of building energy monitoring and management system based on wireless sensor networks. In *2015 Tenth International Conference on Computer Engineering Systems (ICCES)*. IEEE, New York, NY, USA, 230–233. DOI: https://doi.org/10.1109/ICCES.2015.7393051

[3] Adafruit.IO. 2019. Adafruit IO. (2019). https://io.adafruit.com

[4] Amazon Web Services, Inc. 2018. IoT Applications & Solutions | AWS. (Sep 2018). https://aws.amazon.com/iot/

[5] N. Anoush. 2016. Performance trends and alerts with ThingSpeak IoT. https://www.slideshare.net/AnoushNajarian/performance-trends-and-alerts-with-thingspeak-iot. (2016).

[6] blynk. 2019. Blynk. (2019). https://blynk.io/

[7] D. Dobrilovic and S. Zeljko. 2016. Design of open-source platform for introducing Internet of Things in university curricula. In *2016 IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, New York, NY, USA, 273–276. DOI: https://doi.org/10.1109/SACI.2016.7507384

[8] Charalampos Doukas. 2012. *Building Internet of Things with the Arduino*. CreateSpace Independent Publishing Platform, USA.

[9] S. Gangopadhyay and M. K. Mondal. 2016. A wireless framework for environmental monitoring and instant response alert. In *International Conference on Microelectronics, Computing and Communications (MicroCom)*. IEEE, New York, NY, USA. DOI: https://doi.org/10.1109/MicroCom.2016.7522535

[10] Google. 2018. Google Cloud IoT - Fully managed IoT services | Google Cloud. (Aug 2018). https://cloud.google.com/solutions/iot/

[11] Google. 2018. ThingSpeak - Google Trends. (Sep 2018). https://trends.google.com/trends/explore?q=ThingSpeak

[12] U Haque. 2004. Pachube. http://www.pachube.com. (2004).

[13] IFTTT. 2018. IFTTT Homepage. (Sep 2018). https://ifttt.com/

[14] International Organization for Standardization . 2018. ISO 639 Language codes. https://www.iso.org/iso-639-language-codes.html. (2018).

[15] IOBridge. 2018. iobridge/thingspeak. https://github.com/iobridge/thingspeak/. (2018).

[16] Melody Y. Ivory and Marti A. Hearst. 2002. Statistical Profiles of Highly-Rated Web Sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. Association for Computing Machinery, New York, NY, USA, 367–374. DOI: https://doi.org/10.1145/503376.503442

[17] Lab11. 2022. Releases — lab11/thingspeak-scraper. https://github.com/lab11/thingspeak-scraper/releases. (2022).

[18] langdetect. 2018. langdetect: Port of Google's language-detection library to Python. https://github.com/Mimino666/langdetect. (2018).

[19] Luis Leiva. 2012. Automatic web design refinements based on collective user behavior. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. ACM, ACM, New York, NY, USA, 1607–1612.

[20] Kristina Lerman. 2007. User participation in social media: Digg study. In *Proceedings of the 2007 IEEE/WIC/ACM international conferences on web intelligence and intelligent agent technology-workshops*. IEEE Computer Society, IEEE, New York, NY, USA, 255–258.

[21] Marcello A Gómez Maureira, Daan Oldenhof, and Livia Teernstra. 2011. ThingSpeak–an API and Web Service for the Internet of Things. (2011).

[22] Vicki McKinney, Kanghyun Yoon, and Fatemeh "Maria" Zahedi. 2002. The measurement of web-customer satisfaction: An expectation and disconfirmation approach. *Info. Sys. Research* 13, 3 (sep 2002), 296–315. DOI: https://doi.org/10.1287/isre.13.3.296.76

[23] Microsoft. 2018. Azure Internet of Things - IoT for Every Business | Microsoft Azure. (Sep 2018). https://aws.amazon.com/iot/

[24] Microsoft. 2018. Introducing Text Analytics in the Azure ML Marketplace | Machine Learning Blog. https://blogs.technet.microsoft.com/machinelearning/2015/04/08/introducing-text-analytics-in-the-azure-ml-marketplace/. (2018).

[25] Microsoft Azure. 2018. Using Python to call the Text Analytics API - Microsoft Docs. https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/quickstarts/python. (2018).

[26] Ghulam Fiza Mirza, Aqeel Ahmed, Nafeesa Bohra, Sorath Khan, Azam Rafique Memon, and Anum Talpur. 2018. Performance Analysis of Location Based Smart Catastrophe Monitoring System Using WSN. *Wirel. Pers. Commun.* 101, 1 (July 2018), 405–424. DOI: https://doi.org/10.1007/s11277-018-5696-3

[27] N. M. Morshed, G. M. Muid-Ur-Rahman, M. R. Karim, and H. U. Zaman. 2015. Microcontroller based home automation system using Bluetooth, GSM, Wi-Fi and DTMF. In *2015 International Conference on Advances in Electrical Engineering (ICAEE)*. IEEE, New York, NY, USA, 101–104. DOI: https://doi.org/10.1109/ICAEE.2015.7506806

[28] myDevices. 2019. Home. (2019). https://mydevices.com/

[29] Sharmad Pasha. 2016. ThingSpeak based sensing and monitoring system for IoT with Matlab Analysis. *International Journal of New Technology and Research (IJNTR)* 2, 6 (2016), 19–23.

[30] Laura Pravato and Thomas E. Doyle. 2017. IoT for Remote Wireless Electrophysiological Monitoring: Proof of Concept. In *Proceedings of the 27th Annual International Conference on Computer Science andSoftware Engineering (CASCON '17)*. IBM Corp., Riverton, NJ, USA, 254–258. http://dl.acm.org/citation.cfm?id=3172795.3172826

[31] Leonard Richardson. 2007. Beautiful soup documentation. (2007).

[32] Young-Woo Seo and Katia Sycara. 2004. *Text Clustering for Topic Detection*. Technical Report CMU-RI-TR-04-03. Carnegie Mellon University, Pittsburgh, PA.

[33] Ben Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE, IEEE, New York, NY, USA, 336–343.

[34] Sparkfun. 2018. Tutorial: Internet of Things Experiment Guide. https://learn.sparkfun.com/tutorials/internet-of-things-experiment-guide. (2018).

[35] The Mathworks, Inc. 2018. Basic Thingspeak Introduction Using Python: 3 Steps. https://www.instructables.com/id/Basic-Thingspeak-Introduction-Using-Python/. (2018).

[36] The Mathworks, Inc. 2018. How to Buy. https://thingspeak.com/prices/thingspeak_student. (2018).

[37] The Mathworks, Inc. 2018. MATLAB Toolboxes Available on ThingSpeak. https://blogs.mathworks.com/iot/2016/11/05/matlab-toolboxes-on-thingspeak/. (2018).

[38] The Mathworks, Inc. 2018. ThingSpeak - PyPI. https://pypi.org/project/thingspeak/#history. (2018).

[39] The Mathworks, Inc. 2018. ThingSpeak Internet of Things. (Sep 2018). https://thingspeak.com/

[40] Thingful. 2019. Thingful A search engine for the IoT. (2019). https://www.thingful.net

[41] Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L. Littman. 2014. Practical Trigger-Action Programming in the Smart Home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 803–812. DOI: https://doi.org/10.1145/2556288.2557420

[42] Blase Ur, Melwyn Pak Yong Ho, Stephen Brawner, Jiyun Lee, Sarah Mennicken, Noah Picard, Diane Schulze, and Michael L. Littman. 2016. Trigger-Action Programming in the Wild: An Analysis of 200,000 IFTTT Recipes. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3227–3231. DOI: https://doi.org/10.1145/2858036.2858556

[43] Paola Velardi, Alessandro Cucchiarelli, and Michael Petit. 2007. A Taxonomy Learning Method and Its Application to Characterize a Scientific Web Community. *IEEE Transactions on Knowledge and Data Engineering* 19, 2 (2007), 180–191. DOI: https://doi.org/10.1109/TKDE.2007.21

[44] Haoxiang Yu, Jie Hua, and Christine Julien. 2021. Analysis of IFTTT Recipes to Study How Humans Use Internet-of-Things (IoT) Devices. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems (SenSys '21)*. ACM, New York, NY, USA, 537–541. DOI: https://doi.org/10.1145/3485730.3494115