Analysis of Continuous Integration (CI) in Stack Overflow Posts

Shriya Satish
School of Computer Science
Carleton University
Ottawa, Canada
shriyasatish@cmail.carleton.ca

***Abstract*—Continuous Integration (CI) approaches, procedures, and technologies are continuously emerging in response to the rising demand of rapid software delivery and deployment. This necessitates an awareness of contemporary technological advances, as well as developers concerns and perspectives in this field. Continuous Integration (CI) is said to bring a number of advantages to software development, including increased software quality and dependability. However, recent research has identified hurdles, roadblocks, and undesirable practises that characterise its adoption. By mining posts from the Stack Overflow (SoF) community, this project aims to deliver an empirical investigation identifying the software practitioners' viewpoints on the recent trend of Continuous Integration technology.**

Through this empirical study we aim to study the co-existing tags, the dominant topics in Continuous Integration field, the popularity by seeing the view, score and favourite count and check the evolution of Continuous Integration through the years.

***Index Terms*—Continuous Integration, Stack Overflow, Empirical Study, Qualitative Analysis, Topic Modelling.**

I. Introduction

Release engineering is concerned with the operations that occur between the normal development of a software product and its distribution to the end user. Release engineers turn developers' source code into a product suitable for customers' consumption through a number of processes such as code integration from development branches, build compilation, packaging, testing, and signing of the product for release. Continuous Integration is one such release engineering strategy (CI). The Continuous Integration (CI) community has seen a considerable shift toward new approaches for delivering high-quality Software products on a continual basis. This new paradigm is known as Development and Operations (DevOps), and it aims to reduce the software development life cycle by collaborating closely between development and operations teams.

Continuous Integration (CI) is a build process that is automated and runs on dedicated server machines with the goal of finding integration errors as soon as feasible on the go that leads to Continuous Delivery. Significant gains in productivity, customer happiness, and the capacity to ship high-quality goods through quick iterations were observed by industrial enterprises who switched to CI. Many Open Source Software (OSS) contributors have adopted the CI process because of its undeniable advantages, making it one of the most frequently used software engineering approaches [15]. Due to its recent popularity, the number of CI-related queries and users in SoF has increased in recent years, resulting in an increase in developers adopting it and opening up new issues in CI-related activities.

Stack Overflow (SO) is a prominent and popular Question and Answer (Q&A) website where Software Engineers frequently ask questions. Since its inception in 2008, SO has been a host platform for over 18 million questions on various topics of software development. As SO includes a large volume of data based on Q&A that captures developers opinions, it has become a popular source for Software engineering research.

By mining posts from the SO community, the aim is to provide an empirical research analysing practitioners' view, the evolution and the trends associated with Continuous Integration technology. SO is a rich source of data in which different points of view and viewpoints are addressed independent of the organisation. This enables the investigation of trends and an overview of current CI practises. This can be accomplished through an empirical investigation by performing Topic Modelling and utilising Natural Language Processing (NLP) tools and Qualitative Analysis methods to give a better insight into developers and practitioners questions related to CI. By comparing the many topics that coexist with the CI tag, we may uncover evidence and examine the evolution of the CI topics on Stack Overflow and the trend of popularity. We may identify future work to focus on the most popular and problematic CI subjects as a result of this research.

The main motivation is due to the popularity of CI tag in recent years [15], there has been a significant increase in the interest and use of CI frameworks, which has piqued the interest of CI practitioners and software engineering researchers. Identifying the primary subjects of various CI systems across different platforms will aid in identifying the major issues that CI practitioners encounter and providing some potential solutions. The contributions of this study are four-fold by answering three questions that deal with: (i) a finding of top coexisting CI-related tags in SO; (ii) a model for extracting the top dominant topics from discussions in SO; (iii)a manually curated dataset of CI discussions in SO to study the trend and evolution of tags over the years.

In this project report, I will discuss the Related Work, Methodology , Research Questions- their motivation, approach

and results, Threats to Validity, Conclusion and References.

II. Related Work

Many publications have looked into the viewpoints of software engineers by mining software repositories (e.g., Github) and Q&A websites (eg, SO). The Stack Overflow data dump has been widely used in various studies, and the researchers used the Stack Overflow data dump to tackle a variety of issues. Among all the studies, certain researchers focused on Stack Overflow development. Barua et al. [10] evaluated and conducted textual analysis of SO discussions and the correlations and trends of Stack Overflow topics. Some research that use SO data in areas including Deep Learning [8], Mobile [9], Security [7], Big Data [6] and Web development [10]. Existing literature, on the other hand, does not specifically address CI technology in particular and focus more on Microservices [4], Chatbots [3] or Dockerfiles [5] tackles topics that are relatively related.

Through the work of Zahedi et al. [1], the authors try to understand the recent trends in the Continuous Software Engineering (CSE) field by analysing and presenting an empirical study on the CSE tag. Openjaet al. [2] present an empirical study on a broad range of topic related to Release Engineering and perform Topic Modelling techniques as well. They present an extensive work on how the data is retrieved from Stack Overflow dataset, extracted the tags, perform LDA Topic modelling [12] to answer Research Questions to determine the popular topics, the difficult topics, correlation between the popular and difficult questions and determine the type of questions. They perform Topic Modelling to determine the most popular and dominant topics and also do a qualitative analysis to identify challenges and see the evolution of the tag over the years taken in consideration. One of the popular topic in this analysis was "Continuous Integration". The questions have grown from basic general concepts to more technical-specific inquiries, which are more difficult to answer well. Haque et al. [5]delivers a large-scale empirical investigation identifying practitioners' viewpoints on Docker technology on a set of relevant tags and contents, a dataset of Docker-related posts was built. The data was cleaned and processed for analysis. LDA [12]was used to undertake topic modelling, which allowed the domain's dominating subjects to be identified.

When considering the methodology used to undergo analysis Abdellatif et al. [3], Bagherzadeh et al. [6], Han et al. [8], the authors investigate the Stack Overflow website and provide insights on the popular and difficult topics when it comes to Chatbot [3],Big Data [6] and Deep Learing [8] tag and discuss the challenges they face. They provide an insight on how the categories of each topic discussed under the tag evolved over time and also investigate the reasons behind the sudden hike in posts during some periods of time using LDA Topic Modeling [12], Natural Language Understanding (NLU) techniques and qualitative analysis. It provides a good study to be take forward and useful for Researchers and practitioners in the field. This study on CI helps to take inspiration from these work to study on the relevant topics in CI and see the evolution of the tag through the years.

Bandeira et al. [4] worked on dividing microservice tagged articles into three categories: technical, conceptual, and unrelated talks, and then used a topic modelling technique to categorise the conversations based on the first two groups. This study include Continuous Integration as one of the popular topics into consideration. Thus, this project can be taken as a consideration of a future work for this paper.

Zampetti et al. [15] studies conducts an empirical investigation of the problematic behaviours seen by developers using CI. Continuous Integration (CI) is said to bring various advantages to software development, including higher software quality and dependability. However, recent research has identified problems, hurdles, and undesirable practises that characterise its adoption. This study will help to evaluate the evolution of practices based on the challenges faced by practitioners in the field.

Thus with the help of these works, I aim to study on the topic of Continuous Integration and uncover new analysis and findings for developers in the fields practicing it that will provide them with insights on the popularity of the topics and tags and which are more relevant and evolving topic. The work is related but different different from the above studies as in this study, an empirical study is performed on the CI tag related posts on Stack Overflow. In this work 12628 CI related post are extracted, modelled and categorized to understand CI topics that developers are interested in discussing with others in the community, their popularity or the evolution of the posts, implications of such understanding for practice, research and education of CI related activities in the future.

III. Methodology

The purpose of this research is to examine SO posts about CI and characterise them from numerous perspectives. The aim is to look at how the quantity of such tags changes over time, as well as the most popular topics and tags linked with CI. The viewpoint is beneficial for educators who are interested in learning CI problems. They can use this data to develop training materials that introduce these concepts. Researchers might also use the data to create better tools and analytic techniques to assist CI developers during the difficult periods of the ML system development life cycle. The context comprises of SO postings relating to CI that were created between 2008 and 2020 and pertain to various programming languages.

Our study technique consists of the following steps. In the subsequent phases, First, I'll explain the data preprocessing process, and then we'll go through the top tags that coexist with CI. Then, at different workflow stages, we employ a topic model to discover LDA-topics and aggregate the resultant LDA-topics into distinct subject categories. Finally, we expand on the measurements and analyses. Each stage is described in detail in the subsections that follow.

1) Data Collection:

- **Step 1: Download and Retrieve data from Stack Exchange Data Explorer**- To obtain a large number of Stack Overflow questions on CI, use SQL queries to query the official Stack Overflow database at https://data.stackexchange.com/stackoverflow/queries. These queries can only return 50,000 results at a time, and the result of our query mentioned below in Step 2 returns 12,628 posts. This report's time span was set from 2018 to 2020.
- **Step 2: Identify and Extract Continuous Integration Tag**- The Continuous Integration tag is extracted and taken into consideration. Also, the highly relevant tags in context to the study that coexist are taken into consideration. Tag based and Content based filtering can happen to extract this data. RQ1 can be answered using this question. he query is given as follows:

  SELECT * FROM Posts
  WHERE Tags LIKE '%continuous-integration%'
  ORDER BY Posts.CreationDate desc

  The SO dataset contains a large number of question and answer posts, each with its own set of data. The data for a post comprises its identifier, title, body, tags, creation date, view count, score, favourite count, to name a few.
- **Step 3: Extract the posts**- The different posts filtered based on the previous step are extracted and taken as the final data used for this study. 12628 rows of data have been extracted and used for the purpose of this research project. We use the following columns for the dataset: ID, CreationDate, ViewCount, Title, Body, Score, FavouriteCount.

2) Data Preprocessing: - The dataset is further explored in this stage. As SO posts are frequently accompanied by code samples, and the code snippets typically contain programming language syntax and keywords, this may result in poor subject modelling findings and bring noise into future research analysis. Furthermore, because the majority of the source code on Stack Overflow is composed of tiny segments, there is insufficient context to extract significant material from the code snippets (Barua et al. [10]). As a result, code snippets contained in the Stack Overflow dataset's ¡code¿ HTML element is removed .Some preprocessing and cleaning takes place to check if any missing values are present or if any duplicated values need to be removed that may have been created due to the presence of multiple tags.

- **Check missing values:** We check if there are any Missing Values present in any of the columns to avoid any issues while doing the empirical study. FavoriteCounts column had a lot of missing values and were fixed by data preprocessing techniques.
- **Tokenization:** Tokenization is a process that divides and splits large strings of text into smaller chunks, or tokens. Larger portions of text can be tokenized into sentences, then sentences into words, and so on. After a piece of text has been correctly tokenized, additional processing is usually conducted. Text segmentation and lexical analysis are other terms for tokenization.
- **Filtering and Removing Stop Words:** Since the presence of typical English-language stop in a sentence words do not generate relevant subjects (Schutze et al. 2008 [16]), we filter them out using the NLTK stop words corpus (Loper and Bird 2002 [17]).
- **Removing Numbers, Punctuation and other Non-Alphabetic Characters:** It is beneficial to eliminate numbers, punctuation marks, and other non-alphabetic characters from the dataset for improved topic modelling results.
- **Removing HTML Tags and URLs:** Since HTML elements (e.g., ¡p¿, ¡pre¿, and ¡b¿) and URLs are worthless and of no use in the topic modelling process, we delete all HTML tags and URLs from the dataset we have.
- **Stemming:** Stemming is the process of eliminating all forms of affixes from a word, including suffixes, prefixes, infixes, and circumfixes, in order to get a word stem. Using Porter stemmer, it is possible to reduce words to their stemmed representations. 'Programmer,' for example, was abbreviated to 'programme,' while 'configuration,' 'configure,' and 'configured' were all shortened to 'configr.'
- **Lemmatization:** Lemmatization is similar to stemming, however it differs in that it can capture canonical forms based on the lemma of a word. Lemmatization is the process of doing things right by employing a vocabulary and morphological analysis of words, with the purpose of eliminating only inflectional endings and restoring the base or dictionary form of a word, which is known as the lemma.
- **Removal of code snippets:** As Stack Overflow posts frequently include code snippets, which often contain programming language syntax and keywords, this could lead to poor topic modelling findings and bring noise into future analysis. Furthermore, because the majority of the source code on Stack Overflow is made up of tiny segments, there isn't enough context to extract significant content from the code snippets. As a result, we moved the code snippets that were encased in the Stack Overflow dataset's code¿ HTMLtags, where the preprocessing method was the same as in earlier studies [10].

3) LDA Topic Modelling: - Topic modelling is a strong text mining approach for data mining, latent data discovery, and discovering correlations between data and text documents. Latent Dirichlet Allocation (LDA) is one amongst the popular approaches when Topic Modelling is taken into consideration. LDA is a cutting-edge, extensively used topic modelling approach that approximates real-world scenarios by modelling a subject as a group of often co-occurring terms. LDA [14] is used to extract topics from source code and conduct software similarity visualisation. In other words, LDA is utilised as an intuitive way for calculating similarity across source files and

obtaining their respective distributions of each document over topics.LDA is a probabilistic generative model of a corpus. The core notion is that the texts are represented as random mixes across latent themes, with a topic defined by a word distribution. Topics are represented by word probabilities. The terms with the highest probability in each topic often provide a solid indication of what the topic is about. LDA [14] algorithm will run on the dataset by grouping the posts into a set of topics based on the word frequencies and further experiment and investigate on the different topics to answer the proposed Research Questions. This means LDA is probabilistic. After *i* iterations of grouping, the postings are sorted into *K* subjects. A document is a vector of topic probabilities, and a topic is a vector of word probabilities. The most prominent subject is the one with the largest percentage value.

4) Qualitative Analysis: - Here, the evolution of the tags and the trend will be examined so that it gives a visual and qualitative analysis of the evolution of the different topics in Continuous Integration tags.

Fig. 1 show the proposed flow of steps to be followed through this Project.

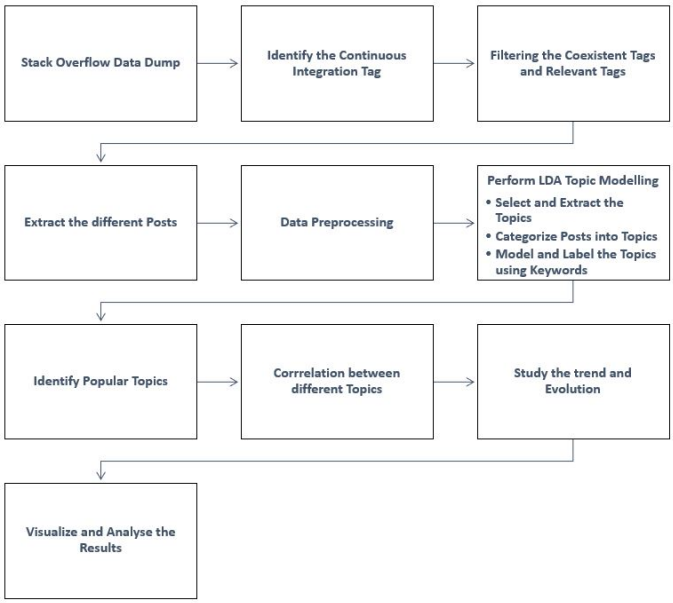


Fig. 1. Proposed Methodology

IV. Research Questions - Motivation, Approach and Results

This section suggests the possible Research Questions, this project will aim to answer:

- **RQ1: What are the common tags that co-exist with Continuous Integration?**
  Motivation: The purpose of this research question is to highlight the prominent tags that co-exist when the CI tag is used.
  Approach: Exploratory Data Analysis is used to find the Number of counts and the total views of each tag from the Tags column. We first split and clean the Tags to separate the different tags and take the unique tags and get the count associated with and accumulate the total views that occur.

	Tag	Counts	Views
0	continuous-integration	12628	29055852
1	jenkins	2754	12043436
2	continuous-deployment	1203	1696022
3	gitlab	1044	1978330
4	git	972	2489072
5	azure-devops	840	1035208
6	teamcity	780	1688846
7	docker	749	1077732
8	github	669	1080979
9	gitlab-ci	592	1287536
10	java	559	1625143
11	hudson	531	3707214
12	maven	454	1392807
13	build	450	1473407
14	travis-ci	441	825500
15	msbuild	422	1345121

Fig. 2. Tags with counts and views

Result: This research question can help you have a better understanding of CI postings. Fig. 2 shows the the tabular form of views and counts of the particular tag associated with CI on Stack Overflow. Continuous-integration tag itself is the most popular tag in this research followed by jenkins, continuous-deployment, gitlab, git, azure-devops,teamcity, docker, github. Fig. 3 shows the visual representation of the count of tags discussed regarding CI on Stack overflow

- **RQ2: What Continuous Integration topics do developers ask questions about?**
  Motivation: The purpose of this research question is to highlight the topics and categories covered by posts about Continuous Integration. The goal of the first research question is to highlight the areas covered by CI-related queries posted on Stack Overflow. This research question

can provide developers with a greater understanding of CI-related concerns and make them aware of various CI topics.

Approach: LDA, a robust topic model, is employed to cluster the CI-related questions [14]. LDA, in general, requires a predefined number of subjects *K*, with variable optimum values of *K* for different tasks. This method may automatically calculate a (near) ideal value of *K* based on the characteristics of a particular issue, allowing LDA to get a superior outcome. In this project, the *K* values considered were 4,5 and 10 and results have been achieved. The number of topics, represented as *K*, is often a user-specified parameter that may be used to adjust the granularity of the identified topics. The value of *K* cannot be too high or too small; too large a value of *K* may result in subject crossover and redundancy, whilst too small a value of *K* may result in coarser-grained topics. We set K range for the corpus to pick the appropriate K value that yields themes with high coherence score. We compute topic coherence for each LDA model using the gensim module CoherenceModel, which implements Roder et al four-stage topic coherence pipeline [18].

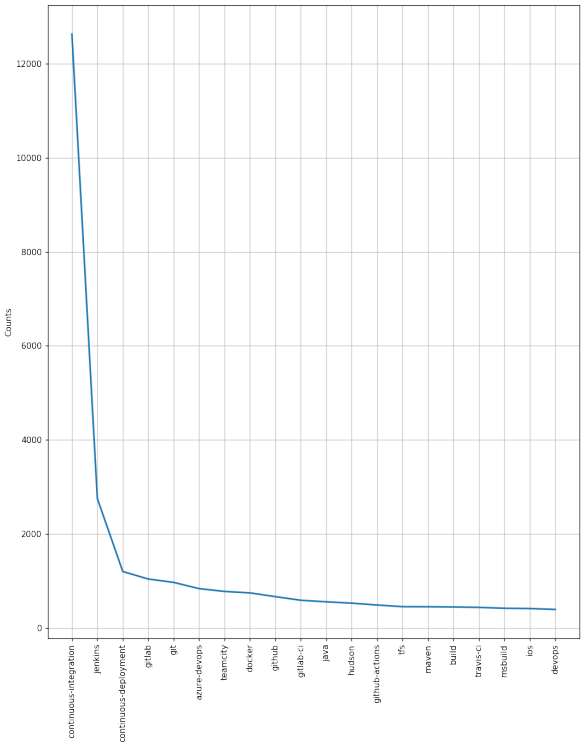


Fig. 3. Top 20 CI tags

Result: This research question can help you have a better understanding of CI postings. LDA topic inference and topic labelling are used to determine the CI-related topics posted by developers on Stack Overflow. Results from a

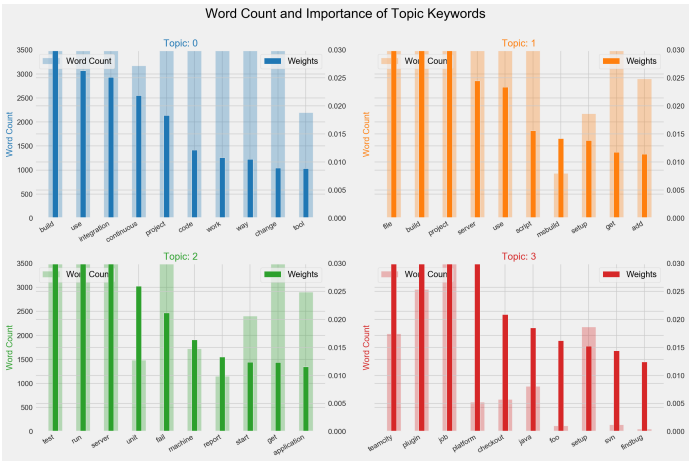


Fig. 4. Word Count and Importance of Topic Keywords

study of the automatically generated topics, the topics manually combined a pair of semantically comparable ones, yielding a total of top 20 topics produced from the Mallet tool. Setting k=4, we get the the topics shown in Table: I, whereas k=20, we get the top 20 topics for CI as shown in Table: II. As k=4 has better coherence value of 0.64 than k=20 (0.36) we consider the k value to be 4 and consider the top 4 topics.

Each document in an LDA model is made up of several topics. However, just one of the issues is usually dominating.So, thus we find the dominant topic in each document as shown in Fig: 7 which gives the contributing percentage of the topic present in the document. The model extracts the dominant subject for each phrase and displays the weight of the topic and the keywords in a beautifully formatted and structured output. We may want to obtain samples of sentences that best reflect a specific topic from time to time. The finest example phrase for each topic is shown in Fig 6.

When dealing with a large number of documents such as this case, we need to know how large the documents are overall and by topic. Fig: 5 represents the distribution of the Document Word Counts by the 4 Dominant Topics.

When it comes to keywords in topics, the importance (known as weights) of the keywords are important. In addition, the frequency with which the terms appear in the texts is worth investigating. The word counts and weights of each keyword are plotted on the same chart in Fig:4. It is sometimes necessary to be cautious of terms that appear in many themes and those whose relative frequency exceeds the weight. Such words are frequently found to be unimportant.

Each word in the document represents one of the four subjects. Let us visualise with colouring each word in the provided documents according to the topic id to which it is assigned. The topic allocated to the document is indicated by the colour of the surrounding rectangle. The sentence chart visualisation for ten documents (Doc 19-

TABLE I
TOP 4 TOPICS OBSERVED

Topic Number	Topic	Topic Words that Represent the Topic	Color Representation
0	Build Process	build use integration continuous project code work way change tool	Blue
1	Server Build	file build project server use script msbuild setup get add	Orange
2	Run Process	test run server unit fail machine report start get application	Green
3	Plugin System	teamcity plugin job platform checkout java foo setup svn findbug	Red

TABLE II
TOP 20 TOPICS OBSERVED

Topic Number	Topic	Topic Words that Represent the Topic
0	Github Repository	repository push git pull develop resource suppose pipeline interface tutorial
1	Master Branch	branch trigger release feature merge master exclude origin filter change
2	Scripting	get set script try number line work execute add follow
3	Plugin	plugin info website jar company monitor community engineer poor clean
4	Continuous Integration	build use project integration continuous code way want work change
5	Version Control	teamcity version control dependency package install information view library download
6	Compile Step	task step compile target reference main publish checkout module dll
7	File System	file directory path enough replace unfortunately include recent css contain
8	Server	server run jenkin start system machine configuration use service user
9	Web Deployment	deploy web environment msbuild deployment production app site simple script
10	Docker	minute image container docker reduce recreate expose compose pod latter
11	Java	report xml java agent wae dispay box com inteal little
12	Error Messages	fail error platform failure runner exception cause occur exit iphone
13	Folder Directory	folder artifact product copy upload desktop addition transfer size optimize
14	Automated Operation	net developer coverage core automatic staging operation normally credential retue
15	Testing	test unit run testing integration result case pass side combination
16	Php	assembly php key phpunit permission validate env split blog bootstrap
17	Cloning	bamboo requirement clone form functionality improve installer address early original
18	Workflow	job full big workflow particular great fast action entire invoke
19	Android	tag android org builder gradle os registry tagging com daemon

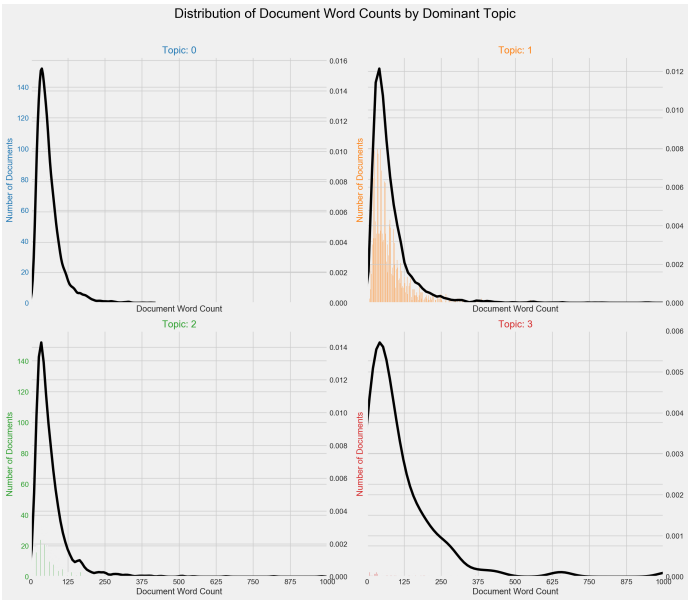
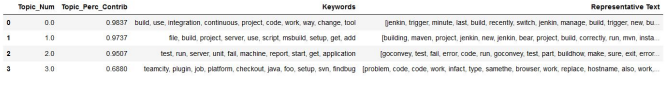
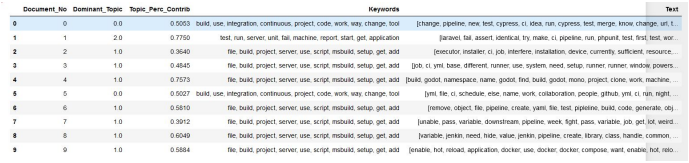


Fig. 5. Distribution of Document Word Count by Dominant Topic



	Topic_Num	Topic_Perc_Contrib	Keywords	Representative Text
0	0.0	0.9837	build, use, integration, continuous, project, code, work, way, change, tool	[jenkin, trigger, minute, last, build, recently, switch, jenkin, manage, build, trigger, new, bu...
1	1.0	0.9737	file, build, project, server, use, script, msbuild, setup, get, add	[building, maven, project, jenkin, new, jenkin, bear, project, build, correctly, run, mvn, insta...
2	2.0	0.9507	test, run, server, unit, fail, machine, report, start, get, application	[goconvey, test, fail, error, code, run, goconvey, test, part, buildhow, make, sure, exit, error...
3	3.0	0.6880	teamcity, plugin, job, platform, checkout, java, foo, setup, svn, findbug	[problem, code, code, work, infact, type, samethe, browser, work, replace, hostname, also, work,...

Fig. 6. Dominant Topic and Contribution Percentage



	Document_No	Dominant_Topic	Topic_Perc_Contrib	Keywords	Text
0	0	0.0	0.5053	build, use, integration, continuous, project, code, work, way, change, tool	[change, pipeline, new, test, cypress, ci, idea, run, cypress, test, merge, know, change, url, t...
1	1	2.0	0.7750	test, run, server, unit, fail, machine, report, start, get, application	[laravel, fail, assert, identical, try, make, ci, pipeline, run, phpunit, test, first, test, wor...
2	2	1.0	0.3640	file, build, project, server, use, script, msbuild, setup, get, add	[executor, installer, ci, job, interfere, installation, device, currently, sufficient, resource...
3	3	1.0	0.4845	file, build, project, server, use, script, msbuild, setup, get, add	[job, ci, yml, base, different, runner, use, system, need, setup, runner, runner, window, powers...
4	4	1.0	0.7573	file, build, project, server, use, script, msbuild, setup, get, add	[build, godot, namespace, name, godot, find, build, godot, mono, project, clone, work, machine, ...
5	5	0.0	0.5027	build, use, integration, continuous, project, code, work, way, change, tool	[yml, file, ci, schedule, else, name, work, collaboration, people, github, yml, ci, run, night, ...
6	6	1.0	0.5810	file, build, project, server, use, script, msbuild, setup, get, add	[remove, object, file, pipeline, create, yaml, file, test, pipleline, build, code, generate, obj...
7	7	1.0	0.3912	file, build, project, server, use, script, msbuild, setup, get, add	[unable, pass, variable, downstream, pipeline, week, fight, pass, variable, job, get, lot, weird...
8	8	1.0	0.6049	file, build, project, server, use, script, msbuild, setup, get, add	[variable, jenkin, need, hide, value, jenkin, pipeline, create, library, class, handle, common, ...
9	9	1.0	0.5884	file, build, project, server, use, script, msbuild, setup, get, add	[enable, hot, reload, application, docker, use, docker, docker, compose, want, enable, hot, relo...

Fig. 7. Most representative sentence for each topic

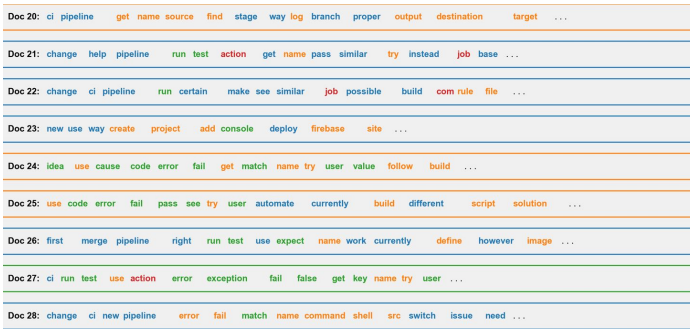


Fig. 8. Sentence Chart Colored by Topic

Doc 28) is represented in Fig:8.

The total number of documents that are attributed to each topic must then be computed. We can accomplish this by plotting two graphs. Firstly, the number of documents for each topic is determined by allocating the document to the topic with the greatest weight in that document. Secondly,

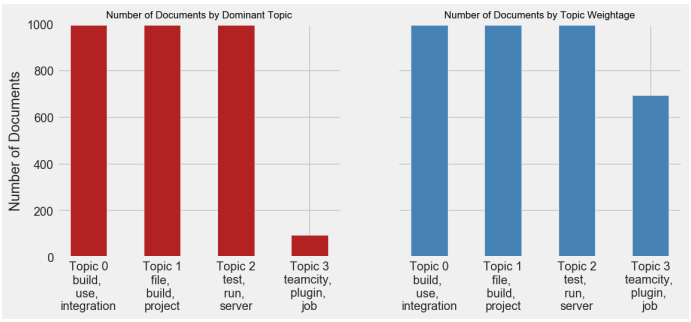


Fig. 9. Most discussed topics in the documents

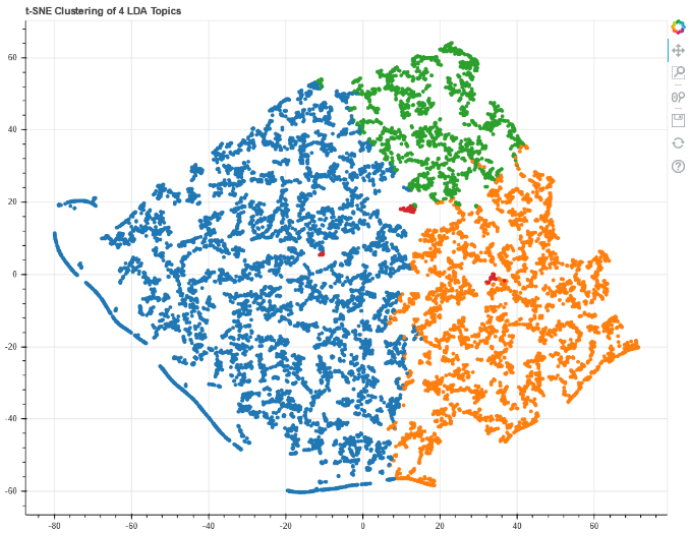


Fig. 10. t-SNE Clustering of 4 LDA Topics

the number of papers for each topic is calculated by adding the real weight contribution of each topic to each document. Fig: 9 shows that even though Topic 0,1,2 are similarly dominant. Topic 0, that is Build Process is the most Dominant Topic.

The t-SNE (t-distributed stochastic neighbour embedding) approach is used to display document clusters in a 2D environment. t-SNE is a non-linear, unsupervised approach that is mostly used for data exploration and visualisation of high-dimensional data. In layman's words, t-SNE provides an impression or intuition of how data is organised in a high-dimensional space. The capacity to retain local structure is the key advantage of t-SNE. This means that points in the high-dimensional data set that are near to one another will likely to be close to one another on the chart. Fig:10 shows the same discussed above and also tells us that Topic 0 is the most dominant topic as well. pyLDAvis is the most widely used and attractive tool for visualising the information contained in a topic model. Fig: 11 shows the pyLDAvis visualization. Each bubble indicates a different topic. The bigger the bubble, the greater the proportion of topics in the corpus that are on that topic. The total frequency of each term in the

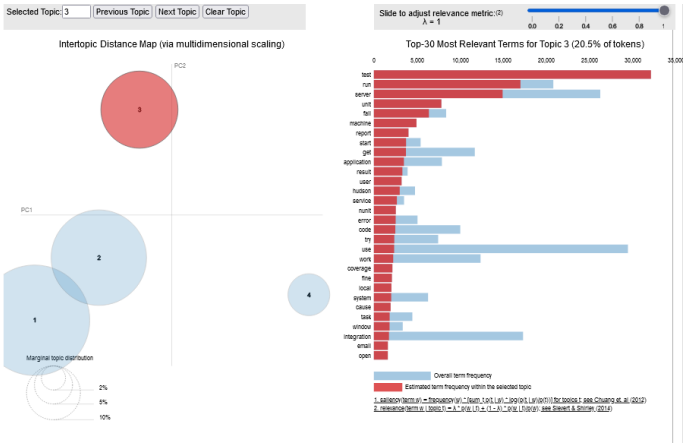


Fig. 11. t-SNE CLustering of 4 LDA Topics

corpus is shown by blue bars. If no topic is chosen, blue bars representing the most commonly used terms will be displayed. The number of times a certain term was created by a given topic is indicated by the red bars.

- **RQ3: What are the characteristics when considering popularity of the identified topics?**-
  Motivation: Following the identification of topics and categories, it can help determine which topics were the most challenging and popular, as well as if there is any association between the two. This research question might aid in the comprehension of CI-related trends and issues. To determine the popularity of a topic, we consider the average number of views, favourite count, and question score. Similarly, the proportion of questions on a topic with no approved answers and the average median time it takes for questions on a topic to obtain accepted responses have been used to evaluate a topic's complexity.
  Approach: To determine the popularity of a topic, we first collect all of the questions related to that topic, and then we use three evaluation metrics based on the metadata of these questions, namely the average number of views, the average number of favourites, and the average score. The property "ViewCount" may be used to directly determine the number of views of a question. The amount of favourites that a question receives may be easily accessed from the question post's characteristic labelled "FavouriteCount." The score of a question may be acquired straight from the property "Score" of the question post. The average number of views is used as the main popularity evaluation metric by default because it measures the average number of developers who view questions about a topic. In theory, a popular question would entice more developers to view it. Nonetheless, the other measures have some reference values for estimating the popularity of the themes.
  Result: Although all of the topics covered in this study are essential in their own right. We concentrated on the most popular CI topics. The Table: III shows the

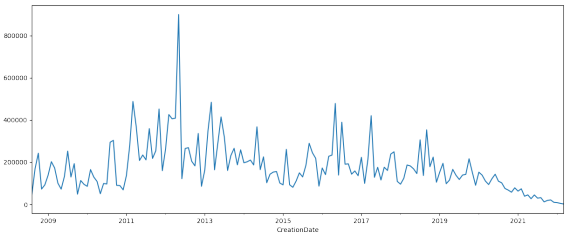


Fig. 12. Evolution Views of CI tags per year

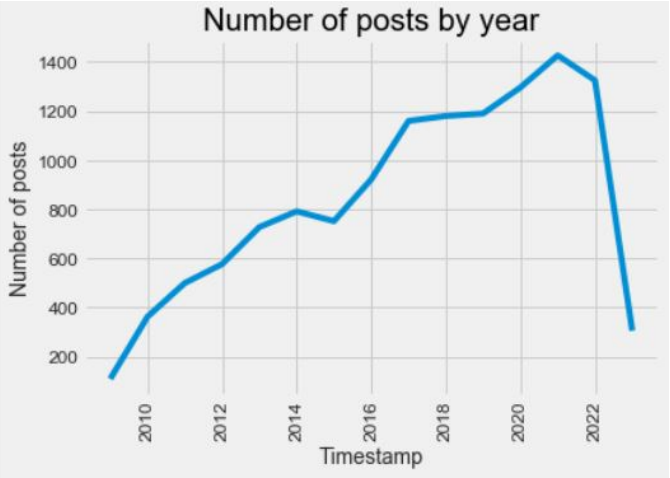


Fig. 13. Evolution of posts of CI tags per year

popularity of the topics in the document in total. It shows that Topic 0: Build Process is the most popular amongst the topics.

- **RQ4: How did the CI Tag evolve through the years?**
  Motivation: We here try to visualize the evolution of CI related posts based on the ViewCount for each post and see the evolution of trend throughout which will help determine the popularity of the tag in general.
  Approach: To consider the trend evolution of all the posts, we first split the CreationDate column to extract the year and try to plot the data of total number of views to the year graph. When considering the Number of Posts in each year, we take the count of the posts in each year into consideration and plot it against the year they were posted.
  Result: We observe from Fig. 12 that 2012 posts are most viewed and they are the most popular times the posts have been answered. When taking the number of questions or posts posted with regards to Continuous Integration, we can observe that 2021 has the highest number of posts where the CI tag was discussed.
  When it comes to the evolution of topics, the red color was coming an issue, so used Purple in place of red. Blue: Topic 0, Pink : Topic 1, Green: Topic 2, Red: Topic 3.

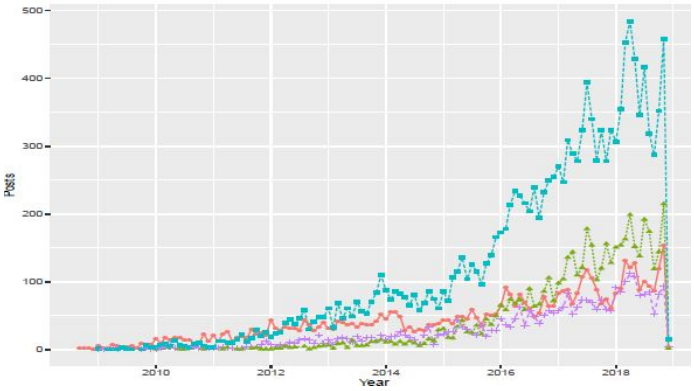


Fig. 14. Evolution Views of CI tags per year

V. Discussions

When studied with related work, Table: IV shows a comparative study with few of the Related Work discussed in Section 2. The comparative study is done with respect to Number of posts, Average ViewCounts, Average FavouriteCouonts, Average Score, Average AnswerCount and Median time taken to answer.

VI. Threats To Validity

There are various threats that might jeopardise the validity of this investigation.

Internal validity threats are caused by errors in the experiments. Personal biases may be apparent in the manual categorising of topics and perception of topics. The interpretation of the topics was done in open-debate as I was the only one who worked on the project. Furthermore, there may have been CI-related discussions that did not use the CI tag and hence were not included in our analysis.Manual labelling, in which we read the question posts to properly map them to the topics and is no way to conduct this operation automatically. Another potential risk arises while determining the appropriate number of topics K and iterations value I.

The fact that we focus on a single website, Stack Overflow, poses an external threat to the validity of our results. Stack Overflow, on the other hand, is presently one of the most popular and largest question and answer websites for software professionals. At the same time, Stack Overflow is a very new site, having launched in 2008, and so does not reflect all of the challenges that web developers have encountered in their development endeavours.

Construct Validity Threats revolve around the metrics being used in this project. This may arise when we determine the popularity of the topic. There may be other better metrics to calculate the Popularity or finding which is a better topic when considered..

VII. Conclusion and Future Work

In this empirical study, I studied the developer discussions on Continuous Integration on the famous Stack Overflow website. The top co-existing tags that occur along with Continuous

TABLE III
POPULARITY OF TOPICS

Topic Number	Topic Name	Average View	Average Favorites	Average Score	Popularity
0	Build Process	12511.5	0.8	1.8	0.421
1	Server	1613.8	0.6	1.6	0.314
2	Run Process	1892.5	0.3	1.2	0.205
3	Plugin System	2780.5	0.2	1.0	0.06

TABLE IV
COMPARISON BETWEEN OTHER STUDIES

Metrics	Chatbot	Mobile	Security	Big Data	Continuous Integration
no. of Posts	3890	16,04,483	94,541	1,25,671	12,628
Avg ViewCounts	512.4	2,300	2,461.10	1,560.40	2,300.9
Avg. Favoritecount	1.6	2.8	3.8	1.9	0.8877
Avg. Score	0.7	2.1	2.7	1.4	3.09
Avg. Answercount	1	1.5	1.6	1.1	1.37
% w/o Answers	67.7	52	48.2	60.3	55
Med. TimeToAnswer	14.8	0.7	0.9	3.3	1.14

Integration tags were observed along with the count and the views of the tag. Jenkins and Continuous Deployment are second and third popular tags. Next using LDA Topic Modelling, the 4 dominant topic were found and visual analysis were done on it. Build Server was the most dominant topic. Lastly the evolution of post and topics were studies based on the count and views. Build Process was evolved the most.

These analysis will help the researchers in the field to get insights on the topics and the keywords. The results may imply that in spite of the growing interest, developers lack proper basic and introductory understanding of CI, and need more information that will help researchers to discuss regarding this.

More topics can be unveiled with better model training and the difficulty of topics and the relation between them can be taken as a future work. We can take other platforms and discussion forums and get better understanding on the topics and understanding the Continuous Integration field.

VIII. GITHUB CODE AND ACKNOWLEDGEMENT

I would like to thank Professor Rabe Abdalkareem for his continuous input in choosing the topic and providing valuable suggestions that will be implemented for the Software Ecosystems (COMP 5900K) as the course project for the term Winter 2022. The code is available at my Github link: https://github.com/shri110797/COMP-5900K–Project–Stack-Overflow-CI-Tag.

REFERENCES

[1] Zahedi M, Rajapakse RN, Babar MA. Mining questions asked about continuous software engineering: A case study of stack overflow. InProceedings of the evaluation and assessment in software engineering (pp. 41-50)2020 Apr 15

[2] Openja M, Adams B, Khomh F. Analysis of modern release engineering topics:–a large-scale study using stackoverflow–. In2020 IEEE international conference on software maintenance and evolution (ICSME) (pp. 104-114).2020 Sep 1 IEEE.

[3] Abdellatif A, Costa D, Badran K, Abdalkareem R, Shihab E. Challenges in chatbot development: A study of stack overflow posts. InProceedings of the 17th International Conference on Mining Software Repositories (pp. 174-185) 2020 Jun 29.

[4] Bandeira A, Medeiros CA, Paixao M, Maia PH. We need to talk about microservices: an analysis from the discussions on stackoverflow. In2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR) (pp. 255-259) 2019 May 25. IEEE.

[5] Haque MU, Iwaya LH, Babar MA. Challenges in docker development: A large-scale study using stack overflow. InProceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (pp. 1-11) 2020 Oct 5.

[6] Bagherzadeh, M. and Khatchadourian, R., 2019, August. 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 (pp. 432-442).

[7] Yang, X.L., Lo, D., Xia, X., Wan, Z.Y. and Sun, J.L., 2016. What security questions do developers ask? a large-scale study of stack overflow posts. Journal of Computer Science and Technology, 31(5), pp.910-924.

[8] Han, J., Shihab, E., Wan, Z., Deng, S. and Xia, X., 2020. What do programmers discuss about deep learning frameworks. Empirical Software Engineering, 25(4), pp.2694-2747.

[9] Linares-Vásquez, M., Dit, B. and Poshyvanyk, D., 2013, May. An exploratory analysis of mobile development issues using stack overflow. In 2013 10th Working Conference on Mining Software Repositories (MSR) (pp. 93-96). IEEE.

[10] Barua, A., Thomas, S.W. and Hassan, A.E., 2014. What are developers talking about? an analysis of topics and trends in stack overflow. Empirical Software Engineering, 19(3), pp.619-654.

[11] Stack Exchange Data Dump: 2022 data https://data.stackexchange.com/stackoverflow/queries

[12] Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y. and Zhao, L., 2019. Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. Multimedia Tools and Applications, 78(11), pp.15169-15211

[13] George A. Miller. 1995. WordNet: a lexical database for English. Commun. ACM 38, 11 (Nov. 1995), 39–41. DOI:https://doi.org/10.1145/219717.

[14] Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3(Jan):993–1022

[15] Zampetti, F., Vassallo, C., Panichella, S., Canfora, G., Gall, H. and Di Penta, M., 2020. An empirical characterization of bad practices in continuous integration. Empirical Software Engineering, 25(2), pp.1095-1135.

[16] Schutze H, Manning CD, Raghavan P (2008) Introduction to information retrieval, vol 39. Cambridge University Press, Cambridge

[17] Loper E, Bird S (2002) Nltk: The natural language toolkit. In: Proceedings of the ACL-02 workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1, Association for Computational Linguistics, pp 63–70

[18] Both A, Hinneburg A (2015) Exploring the space of topic coherence

measures. In: 8th ACM international conference on web search and data mining, pp 399–408