**MINING STACKOVERFLOW QUESTIONS FOR SOFTWARE ENERGY CONSUMPTION USING NATURAL LANGUAGE PROCESSING**

**Abstract**

Transforming into green software engineering involves considering decisions on how to reduce energy consumption of software. However, this has been a secondary concern for much of the last decade. Conventionally speaking, consumption of power was correlated directly to electrical and computer engineering and was thought to be hardware optimization of embedded systems. Though GPUs and other hardware applications continue to be used to regulate power consumption, more responsibility is being shifted to developers of both OS and other applications. Proper documentation of the development of “green” applications and their challenges needs more focus, as such information will be of great value to designers of OS to enhance their API in catering to the challenges related to energy faced by developers, and they, in turn, can use it to formulate more applications that are energy-efficient. Most of the work in the literature help developers to form a prototype of applications for energy consumption and attempt to streamline energy with the help of software repositories. Among these repositories, Q&A websites, such as StackOverflow, are valuable as it gives a comprehensive idea of issues faced by developers and users that are related to energy. Moreover, a number of studies related to energy published in StackOverflow revealed that programmers needed clarifications in many areas of energy consumption. In order to gain factual evidence and appropriate insight on this issue, we analysed programmers discussions to evaluate their knowledge on consumption and efficiency of software energy. The primary source of data utilized in this study is StackOverflow, and the data was curated with the help of NLP (Natural Language Processing) in order to pinpoint relevant queries that are related to the consumption of energy by software.

*Keywords: Energy Consumption, Mining, StackOverflow, Natural Language Processing*

1. **INTRODUCTION**

Over the past ten years, there has been a sudden change in the manner in which computing devices are used by individuals, as well as enterprises. Even the common man has started using computing platforms such as smartphones and watches, tablets, and the way he interacts with software. Even immobile platforms such as datacenters and desktop environments have become very common. Most of the computing devices used by the common man are battery-driven, and thus even a small task that is well-optimized such as texting, consumes energy [1]. Simultaneously, software that is poorly optimized might lead to a waste of energy and drain the battery in a much faster manner. As a result, it is very important to formulate solutions that are highly efficient, have more authentic results, and are quite innovative with regard to the OS, runtime systems, and hardware/architecture.

Formerly, the assumption was that hardware use is the main source of energy consumption, and so concentration should be paid to tasks that consume less energy. Recently, it has been found that running software can use up as much energy as hardware and thus are responsible for the consumption or conservation of energy. Over the last decade, many studies have been conducted to formulate a green software design as a concern of core development in order to enhance the energy conservation of software systems overall. Most of these studies aim to observe energy usage from higher layers of computer stack and application software [2]. These approaches also focus on complementing the prior formed OS/hardware-centric applications so that they are not discarded at the application level while improvising the software applications. Thus both qualitative and quantitative studies [3. 4, 5, 6] have conducted in-depth surveys on the developer’s knowledge quotient of the energy consumption of software.

The efficiency of energy can be influenced by the programmers’ knowledge and their choice of design [1]. So, it is crucial that the software developers should analyze and apply techniques that can help in conserving energy while tackling real-world problems. For instance, the software developers/programmers should come under how energy ‘behaves’ at various levels and design prudently to improve the use of energy effectively [18]. Through extensive studies have been done on software energy conservation, and a number of remedies have been suggested, the perspective of programmers on the consumption of energy has not been focused much. There are no clear answers to common and simple questions related to software types and energy usage, or requirements and energy usage.

The focus of this work is to get better clarity on the view of software engineers on the consumption of energy by software and the manner in which it is being dealt with by them. This issue was studied and researched by many researchers, and a qualitative report was published and summarized based on the experiences of skilled developers [2, 7, 8]. NLP helps in identifying sentiments, topics, location of entities in sentences, and category of blog/article Text mining involves exploration of huge textual data and location of patterns which includes identifying frequent words, sentence length, and absence or presence of specific words. Co-occurring keywords are identified by NLP in order to summarize the large collection of textual information. This helps to discover hidden topics in a document, annotate these topics with the documents, and organize an increased amount of unstructured data. NLP helps in mining appropriate queries that are related to the consumption of software energy [12]. The rest of the paper is organized into 4 sections, presenting the related works, methodology, findings and conclusion.

1. **Related Works**

A qualitative study was conducted by Ournami et al., [2] in an industrial context, where they performed an in-depth analysis of the interviews of 10 experienced developers and summarized it into a set of implications which gave a better idea about the knowledge of developers in the following areas and is listed down as follows:

i. a proper clarity on the developer's idea on the consumption of software energy and designing green software;

ii. the barriers that prevent the adoption of this concept and

iii. the tools and support required by the developers from the company.

The simulation of results gave a better perspective of what is expected from the developers, tool creators, researchers, and the management of the company, in order to prioritize software design for the development of green software.

A survey was conducted on 122 programmers by Pang & Hindle [7], which concluded that the programmers needed proper training on energy consumption, as they lacked the basic knowledge on energy efficiency, were not aware of the practices involved in reducing software energy consumption, and moreover, they were not sure of the amount of energy consumed by software. An empirical study conducted by Manotas et al. [8] gave better clarity on the perspective of practitioners on energy consumption while designing, constructing, and testing software. They concluded their findings based on a 1uantitative survey targeted on 464 practitioners from IBM, Google, ABB, and Microsoft, which was supported and motivated by 18 in-depth interviews with employees from Microsoft. At the end of the study, they suggested contextualizing the existing green software research and gave better directions for researchers while developing tools and strategies to improve green usage in their applications.

Generally, developers depend upon web-based online platforms for the developer community to resolve the issues at hand, as such platforms provide an opportunity for the developers to interact with other programmers and developers in finding a solution for their issues. Stack Overflow is a community website and is a popular platform where queries are answered by appropriate experts and community members. Stack Overflow has gained popularity in the programming industry due to the esteemed and knowledgeable members who participate in providing valuable information to programmers and developers. It also provides new insights and ideas for developers and acts as a platform for discussion, and updates them on changing trends and needs [9].

Stack Overflow conducted a qualitative study based on Q&A to identify the area of energy consumption that the developers think is significant and the solutions suggested by them to enhance energy efficiency [5. 10, 11]. Pinto et al., [5] proposed the first empirical study to understand the views of application-oriented programmers on software energy consumption, where they used data from Stack Overflow as the primary source and curated samples carefully from more than 300 questions and 550 answers, where overall 800 users were analyzed. From their simulation, it was found that though energy consumption was a popular area of analysis, the answers cannot be authenticated qualitatively. Moreover, the observation was that some of the answers were faulty or ambiguous. According to the study, five areas were identified as the most important, and questions were framed in those areas, which included General Knowledge, Measurement, Code Design, Noise, and context Specific. Code modification is the domain that received more focus, and more queries were from that area. At the end of the study, seven major causes for energy consumption were identified, which ranged from background activities to synchronization.

An empirical study was conducted by Malik et al. [10]. Who explored a sample size of 5009 StackOverflow questions based on energy and analyzed 1000 android-related questions manually and studied the issues faced by developers, and their API was also analyzed. The study critically analyzed the concerns of developers on energy-related issues, which focused mainly on sensor and radio utilization and improper implementation. Another critical study was presented by Zaheidi et al., [11], who explored CSE from the perspective of the practitioners by mining the discussions from Q&A websites. 12,989 questions were analyzed by authors, and the answers were documented in StackOverflow. Topics were streamlined so that the dominant topic in the domain was derived. Also, an analysis was done qualitatively to discuss the significant challenges faced, and the conclusion of the study was that the queries to these websites were increasing sharply and technology was the domain of interest, and most of the questions needed thorough expert interpretation.

A number of previous studies with the energy-related questions on StackOverflow revealed that a number of questions were posed by programmers. In order to obtain authentic evidence the insight into this issue, programmers surveyed and gauged the knowledge of consumption and efficiency of software energy. StackOverflow was used as the primary data source, and the data obtained was curated by NLP in order to obtain appropriate answers to the queries posed on software energy consumption and, more specifically, some of the questions that needed answers to include:

RQ1 What are the questions related to software energy consumption that have been raised on Stackoverflow?RQ2 What is the dominant topic of discussion related to energy in StackOverflow?

RQ3 What are the common solutions suggested to software energy consumption to the raised questions/issues?

1. **METHODOLOGY**

Data from StackOverflow was used for our study, which is a popular and efficient Q&A website. This is considered one of the most prominent forums in the arena of software development and is used in software engineering studies. The popularity of this website is proven by the fact that as of March 2021, there are 14 million registered users, and this website has received more than 21 million queries and 31 million answers. Moreover, the data is easily accessible through Stack Exchange Data Explorer, which is an open-source tool that is used to run random SQL queries as opposed to public data from the Exchange network.

Another mode of identifying appropriate answers for the queries is through ‘tags’ that are associated with the questions. Though tag act as a useful navigator but the disadvantage with tag is its inconsistency which can lead to ambiguity due to synonyms, hyponyms, acronyms, and spelling variations, which might lead to the irrelevant text content of the user post and, at times these inconsistencies can lead to a condition called ‘tag explosion’ where certain tags become overused.

The flow of the investigation is shown in Figure 1.

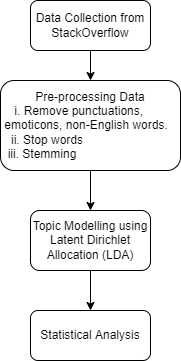


Figure 1. Steps in the Investigation

This paper focuses on questions that are related to software, and terms including “energy efficiency” and “energy consumption” are used for data collection. In the initial stage, the questions, answers tags, and other metadata were extracted from StackOverflow dump. This data is filtered to find the questions related to energy consumption. The tags from these questions are used to extract the questions from the StackOverflow. The data explorer used is StackExchange, and data collection is done using SQL queries that are related to tag and date. Some tags that are used include power management, power-saving, cpu-usage, consumption, energy, energy efficiency, power efficient, power consumption, cpu-architecture, cuda and battery. The duration of the study was from 01-01-2014 to 31-03-2022. Appendix I shows the partial sample of data collected with the help of Stack Exchange Data Explorer. The data is obviously semi-structured data with unstructured questions and answers with standard fields such as ID, CreationDate, Score, ViewCount, Title, and Tags for filtering questions. As tags cannot be used for an answer or comment type openly, most of the content that is generated is untagged. In case of misuse of tags or inappropriate tagging, the posts’ content might not be represented by tags [11].

Raw data should be restructured and reprocessed to remove redundancy, inconsistency, and irregularity, in order to improve the efficiency and quality of the process of classification. This process is carried on by the removal of words that are repetitive, appropriate punctuations, and removal of non-English characters, which might improve the proficiency and adeptness of the data. Tokenizing, re-tweets, hashtags, and URLs should also be removed.

Stop words such as ‘and’, ‘are’, ‘this’, and so on, which occur frequently but do not contribute to the content or context of the textual document, should be removed, though they are essential for the formation of a meaningful sentence. They might not attribute to the classification of the document. With regards to textual sources, this stop word list is inconsistent and difficult, but this process decreases the text data and enhances the performance of the system.

The next important aspect is the ‘stemming’ of words. Stemming algorithms can reduce the grammatical forms in the morphological structure of the language. By stemming, a work is taken to its root level. Various grammatical forms or word forms, such as an adjective, adverb, noun, verb, etc., are taken to their root form. Te [13]. Technically speaking, stemming can be considered as the conflation of a number of forms of words into a single word. To illustrate, the terms close, closed, closed, and closing can be stemmed to close, which is the root form of the word. Stemming is usually done through the removal of attached prefixes and/or suffixes from the index term. As the stem of the term is representative of a broader concept than the original term, the number of retrieved instances increased eventually. As the meaning is the same, but the word form is different, it is essential to identify every word with its base form. The basic idea is to reduce the total number of distinct terms in a document, which will reduce the processing time of the final output.

Another NLP technique is topic modeling, which can extract topics automatically out of the corpus of textual data. Interpretable, semantically consistent topics are generated by topic models, which can be represented through the listing of the most probable words that describe every topic. A topic model is defined as a generative model for documents, as this can help the generation of documents through a probabilistic procedure. During the compilation of a new document, a distribution is selected over topics. Subsequently, for every word, a topic is chosen randomly for every topic based on the distribution, and a word is chosen from the topic. The set of topics can be inferred through standard statistical techniques that are responsible for generating document collection, thus reversing the modelled process of authoring [14].

A popular modelling technique is Latent Dirichlet Allocation (LDA) which helps in generating topics from the corpus of data. It is used commonly in a number of domains, including software mining repository studies. This prototype provides flexibility in topic modelling through the various possibilities of treating a document as a member of multiple topics and in treating a topic as a mixture of words that could lead to a number of topics. According to StackOverflow, a document can be defined as a single thread including a question having a title and body and the appropriate answer. LDA can be considered as a generative probabilistic prototype of a corpus. The basic idea behind this is that the documents can be represented as an arbitrary mixture of latent topics, where the attributes of a topic can be authenticated through the distribution of words. A topic is represented by LDA through word probabilities. In each topic, the word that has the highest probability usually gives a better idea of the probability of the word in LDA.

A corpus is modelled through an unsupervised generative probabilistic method and is considered the most commonly used modeling technique. According to LDA, every document is considered a probabilistic distribution over latent topics, and all documents share a common Dirichlet. In the LDA model, each latent topic is also represented through the probabilistic distribution of words, and there is a common Dirichlet for word distribution prior also. Given a corpus D consisting of M documents, with document d having Nd words (d ∈ 1,..., M), LDA models D according to the following generative process [16]:

(a) Choose a multinomial distribution ψt for topic t (t ∈{1,..., T}) from a Dirichlet distribution with parameter β

(b) Choose a multinomial distribution θd for document d (d ∈{1,..., M}) from a Dirichlet distribution with parameter α.

(c) For a word wn (n ∈{1,..., Nd }) in document d

**3. FINDINGS**

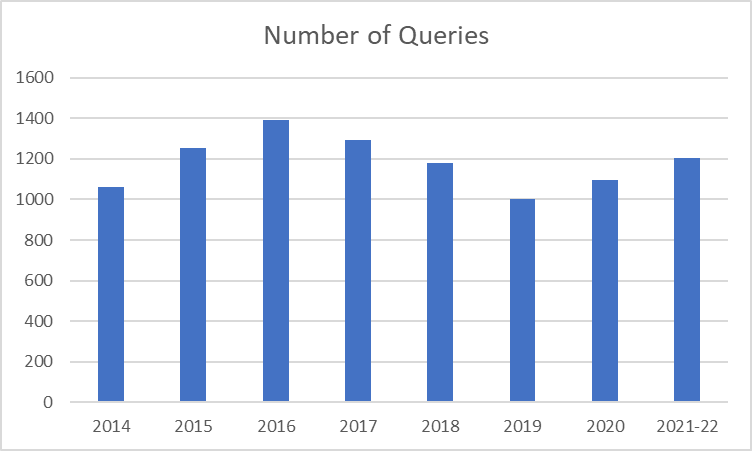
This section presents our findings from analyzing the data collected from StackOverflow. At a high-level, we are interested in answering three main research questions. For answering RQ1, the questions related to software energy consumption addressed in StackOverflow is investigated. To address this, we aimed to connect the score of the questions (queries) and the number of answers related to energy consumption. To answer the RQ2, we used the tags of the queries to find the dominant topic of discussion related to energy in StackOverflow. Also, Topics generated through LDA helps in identifying the topics. Finally, for RQ3, common solutions suggested to software energy consumption to the raised questions/issues is found using the type of devices and solutions offered. The text content of user posts from the Stack Overflow website used to find the technology trends over time, discussion topics among developers.

**3.1 Effectiveness of Answers**

From the post from 01-01-2014 to 31-03-2022, the number of queries related to energy efficiency are 9484. Table 1 gives the breakup of number of queries yearwise. It is observed that the number of queries is increasing.

**Table 1: Number of queries related to Power**

|  |  |
| --- | --- |
| Year | Number of queries |
| 2014 | 1062 |
| 2015 | 1253 |
| 2016 | 1392 |
| 2017 | 1292 |
| 2018 | 1182 |
| 2019 | 1000 |
| 2020 | 1098 |
| 2021-22 | 1205 |
| Total | 9484 |

 **Figure 1: Number of queries related to Power**

The quality of each post according to users is collaboratively evaluated using a voting system. Each question or answer can receive up-votes or down-votes from users, with the sum of votes (up-votes minus down-votes) acting as its overall voting score. The votes awarded to a user’s posts is accumulated in their ‘reputation’, another type of score associated with individual users and intended to identify expert users [17].

The higher the score the better the answers. It is observed from Table 2 that the score range (1 to 5) comprises of 90.53%, (6 to 10) 6.09% and (>10) 3.48%.

Chart, bar chart

Description automatically generated**Table 2: Score range**

|  |  |  |  |
| --- | --- | --- | --- |
| Year\Score | 1 to 5 | 6 to 10 | >10 |
| 2014 | 992 | 60 | 10 |
| 2015 | 1151 | 77 | 25 |
| 2016 | 1280 | 85 | 27 |
| 2017 | 1167 | 90 | 35 |
| 2018 | 1069 | 66 | 47 |
| 2019 | 905 | 55 | 40 |
| 2020 | 976 | 64 | 58 |
| 2021-22 | 1036 | 81 | 88 |
|  | 8576 | 578 | 330 |

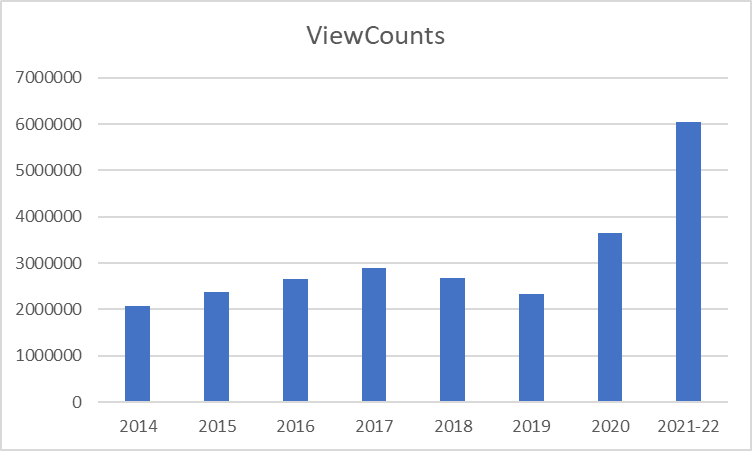
**Figure 2: Score Range**

The maximum score range are above 200 as seen in queries [[1]](#footnote-1),[[2]](#footnote-2), [[3]](#footnote-3), [[4]](#footnote-4).

Table 3 shows the ViewCount for queries. The ViewCount refers to the number of users who viewed the answers. Though the number of queries are more or less in the same range, the views are much higher in the last two to three years showing an increase in interest in energy efficiency.

Table 4 shows the total AnswerCount. Each question can have any number of answers or responses. The answers can reflect different methods to handle a query. This gives an idea on how a developer is resolving the energy consumption issue.

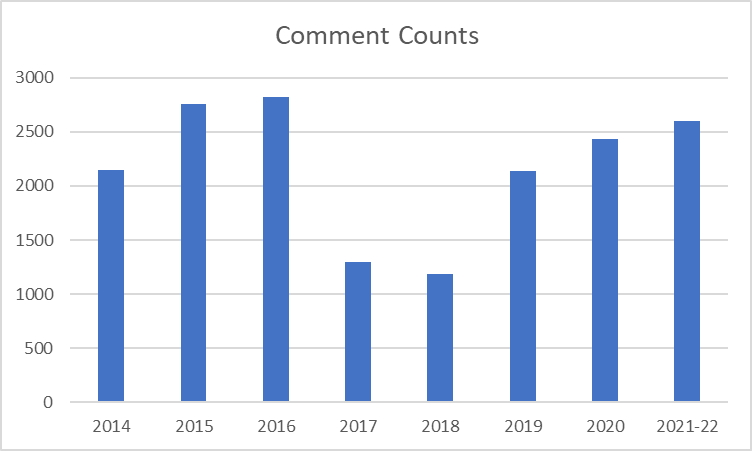
Table 5 shows the total Comments. The comments are not answers to the question but an opinion whether the said answer or solution was effective.

**Table 3: Number of Queries Vs. ViewCount**

|  |  |  |
| --- | --- | --- |
| Year | Number of Queries | ViewCount |
| 2014 | 1062 | 2078425 |
| 2015 | 1253 | 2382159 |
| 2016 | 1392 | 2653435 |
| 2017 | 1292 | 2890438 |
| 2018 | 1182 | 2682463 |
| 2019 | 1000 | 2324758 |
| 2020 | 1098 | 3644762 |
| 2021-22 | 1205 | 6045704 |
| Total | 9484 | 24702144 |

**Figure 3: Number of Queries Vs. ViewCount**

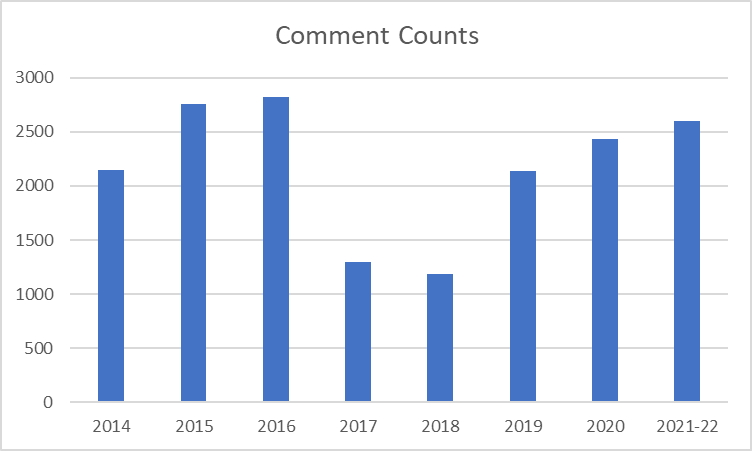
Some of the queries with maximum view count[[5]](#footnote-5), [[6]](#footnote-6).

**Table 4: Number Answer Count**

|  |  |  |
| --- | --- | --- |
| Year | Number of Queries | Answer Count |
| 2014 | 1062 | 1182 |
| 2015 | 1253 | 1322 |
| 2016 | 1392 | 1534 |
| 2017 | 1292 | 1360 |
| 2018 | 1182 | 1294 |
| 2019 | 1000 | 1130 |
| 2020 | 1098 | 1282 |
| 2021-22 | 1205 | 1666 |
| Total | 9484 | 10770 |

**Figure 4: Number Answer Count**

Some of the queries with maximum answer count[[7]](#footnote-7), [[8]](#footnote-8).

**Table 5: Number Comment Count**

|  |  |
| --- | --- |
| **Year** | **Comment Count** |
| 2014 | 2150 |
| 2015 | 2759 |
| 2016 | 2818 |
| 2017 | 1292 |
| 2018 | 1182 |
| 2019 | 2139 |
| 2020 | 2431 |
| 2021-22 | 2603 |
| Total | 17374 |

**Figure 5:Number Comment Count**

Some of the queries with maximum comment count[[9]](#footnote-9), [[10]](#footnote-10), [[11]](#footnote-11).

**3.2 Dominant topics related Energy**

Another key mechanism on the Stack Overflow site is the use of tags to identify the content or theme of each post. When a user asks a question, the platform prompts them to add a small number of content tags (at least one and at most five). Total 480 tags were used in the 9484 posts. Figure shows the word cloud of all tags. The dominant words in tags and title were found using statistical methods.

Text

Description automatically generated**Figure 6: Word Cloud of Tags**

**Table 6:Tag Count**

|  |  |
| --- | --- |
| **Popular tags** | **Count** |
| bluetooth | 6711 |
| energy | 4210 |
| cuda | 3874 |
| android | 2371 |
| c++ | 1272 |
| ios | 988 |
| cpu-usage | 891 |
| gpu | 835 |
| java | 492 |
| battery | 441 |
| nvidia | 364 |
| memory | 360 |
| linux | 316 |
| windows | 295 |
| parallel-processing | 145 |

**Table 7: Dominant Words in Title**

|  |  |
| --- | --- |
| **Title** | **Count** |
| Cuda | 2581 |
| bluetooth | 1485 |
| android | 1308 |
| cpu-usage | 762 |
| memory | 516 |
| gpu | 512 |
| ios | 484 |
| Energy | 369 |
| battery | 355 |
| windows | 237 |
| c++ | 167 |
| nvidia | 144 |
| parallel-processing | 111 |
| linux | 106 |
| Power management | 101 |
| java | 99 |
| Consumption | 74 |
| efficiency | 49 |

A picture containing text

Description automatically generated

**Figure 7: Word Cloud of Dominant Words in Title**

**3.3 Other findings**

The major findings from the data are that the number of queries related to power and its management is 9484. The summary of the finding is given in table 7.

**Table 8: Summary of Findings**

|  |  |
| --- | --- |
| **Parameter** | **Findings** |
| Number of queries related to power | 9484 |
| Score range | 1 to 441 |
| Number of Queries | 9484 |
| Number of Tags | 480 |
| Number of ViewCount | 24702144 |
| ViewCount Range | 11 to 515430 |
| Number of Answer | 10770 |
| Number of Comment | 20277 |

**4 IMPLICATIONS/CONCLUSION**

For RQ1, ‘What are the questions related to software energy consumption that has been raised on Stackoverflow?’. The findings that energy-usage requirements are often stated in terms other than energy usage suggests that energy requirements are difficult to specify directly. The findings that developers believe that general patterns lead to good or bad energy usage and high-level designs are impacted by energy usage concerns more frequently than low-level designs.

For RQ2, ‘What is the dominant topic of discussion related to energy in StackOverflow?’. Findings show that the dominant topic include software design. Empirical studies of the impacts of design needs to be thoroughly investigated. While there has been a significant amount of work in understanding how changes made by developers impact energy usage. Programming languages or language features help developers during the development of energy-efficient applications.

For RQ3, ‘What are the common solutions suggested to software energy consumption to the raised questions/issues?’. Findings show that with the development of mobile devices, the energy consumption issues now are varied to include the mobile devices. Android platforms get equal attention compared to the computer systems. It is seen that the software engineering can have a significant impact on energy conservation in different devices. In recent years, there has been an increased focus on developing more energy-efficient software and algorithms, as well as designing devices that use less power. Some of the common solutions suggested were: Optimizing code for efficiency and reducing resource usage, for example, by avoiding costly operations and data structures, minimizing the amount of time the software spends waiting for input or for other events, reducing the frequency with which the software runs, turning off or reducing the intensity of background processes, such as network and disk activity, that consume energy even when the software is idle, using hardware-level power management features, optimizing graphics performance. Overall, the impact of software engineering on energy conservation in devices has likely been positive over the past several years and is expected to continue to improve in the future.

The finding is that the developers believe that they could learn how to improve energy efficiency in several ways. The want to learn is apparent in the comments and indicates that they use the knowhow of other developer, tools, example codes, documentation etc to improve their knowledge.

Mining StackOverflow questions for software energy consumption using NLP helps to identify common patterns and trends in the way that developers discuss software energy consumption, as well as to identify specific questions and answers that are relevant to the topic. In this work, keywords were used to search through the text of questions and answers on StackOverflow to identify relevant posts. Overall, using NLP to mine StackOverflow questions for software energy consumption provides insights into the way that developers think about and discuss this topic, and can help to identify specific questions and answers that are relevant to the topic.

**References**

1. Pinto, G., & Castor, F. (2017). Energy efficiency: a new concern for application software developers. Communications of the ACM, 60(12), 68-75.
2. Ournani, Z., Rouvoy, R., Rust, P., & Penhoat, J. (2020, October). On reducing the energy consumption of software: From hurdles to requirements. In *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (pp. 1-12).
3. Ardito, L., Coppola, R., Morisio, M., & Torchiano, M. (2019). Methodological guidelines for measuring energy consumption of software applications. *Scientific Programming*, *2019*.
4. Pereira, R., Couto, M., Ribeiro, F., Rua, R., Cunha, J., Fernandes, J. P., & Saraiva, J. (2017, October). Energy efficiency across programming languages: how do energy, time, and memory relate?. In *Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering* (pp. 256-267).
5. Pinto, G., Castor, F., & Liu, Y. D. (2014, May). Mining questions about software energy consumption. In *Proceedings of the 11th Working Conference on Mining Software Repositories* (pp. 22-31).
6. Siegmund, N., Dorn, J., Weber, M., Kaltenecker, C., & Apel, S. (2022). Green Configuration: Can Artificial Intelligence Help Reduce Energy Consumption of Configurable Software Systems?. *Computer*, *55*(3), 74-81.
7. Pang, C., Hindle, A., Adams, B., & Hassan, A. E. (2015). What do programmers know about software energy consumption?. IEEE Software, 33(3), 83-89.
8. Manotas, I., Bird, C., Zhang, R., Shepherd, D., Jaspan, C., Sadowski, C., ... & Clause, J. (2016, May). An empirical study of practitioners' perspectives on green software engineering. In 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE) (pp. 237-248). IEEE.
9. Shukla, S. (2020). The curious case of posts on Stack Overflow.
10. Malik, H., Zhao, P., & Godfrey, M. (2015, May). Going green: An exploratory analysis of energy-related questions. In 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories (pp. 418-421). IEEE.
11. Zahedi, M., Rajapakse, R. N., & Babar, M. A. (2020). Mining questions asked about continuous software engineering: A case study of stack overflow. In *Proceedings of the evaluation and assessment in software engineering* (pp. 41-50).
12. Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544-551.
13. Jivani, A. G. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, *2*(6), 1930-1938.
14. Uys, J. W., Du Preez, N. D., & Uys, E. W. (2008, July). Leveraging unstructured information using topic modelling. In *PICMET'08-2008 Portland International Conference on Management of Engineering & Technology* (pp. 955-961). IEEE.
15. Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., & Zhao, L. (2019). Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, *78*(11), 15169-15211.
16. Zhang, Y., Chen, M., Huang, D., Wu, D., & Li, Y. (2017). iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization. *Future Generation Computer Systems*, *66*, 30-35.
17. Moutidis, I., & Williams, H. T. (2021). Community evolution on stack overflow. *Plos one*, *16*(6), e0253010.
18. Ingeno, J. (2018). Software Architect’s Handbook: Become a successful software architect by implementing effective architecture concepts. Packt Publishing Ltd.

**APPENDIX 1**

**Links**

The data was collected from the official StackOverflow site. The link of the same is:

<https://data.stackexchange.com/>

The queries formed as follows for required keywords.

select \* from Posts where PostTypeId = 1 and Tags like '%keyword%'

Screenshots of questions.

1. <https://stackoverflow.com/questions/31326015>

Graphical user interface, text, application, email

Description automatically generated

1. <https://stackoverflow.com/questions/50622525>

Graphical user interface, text, email, website

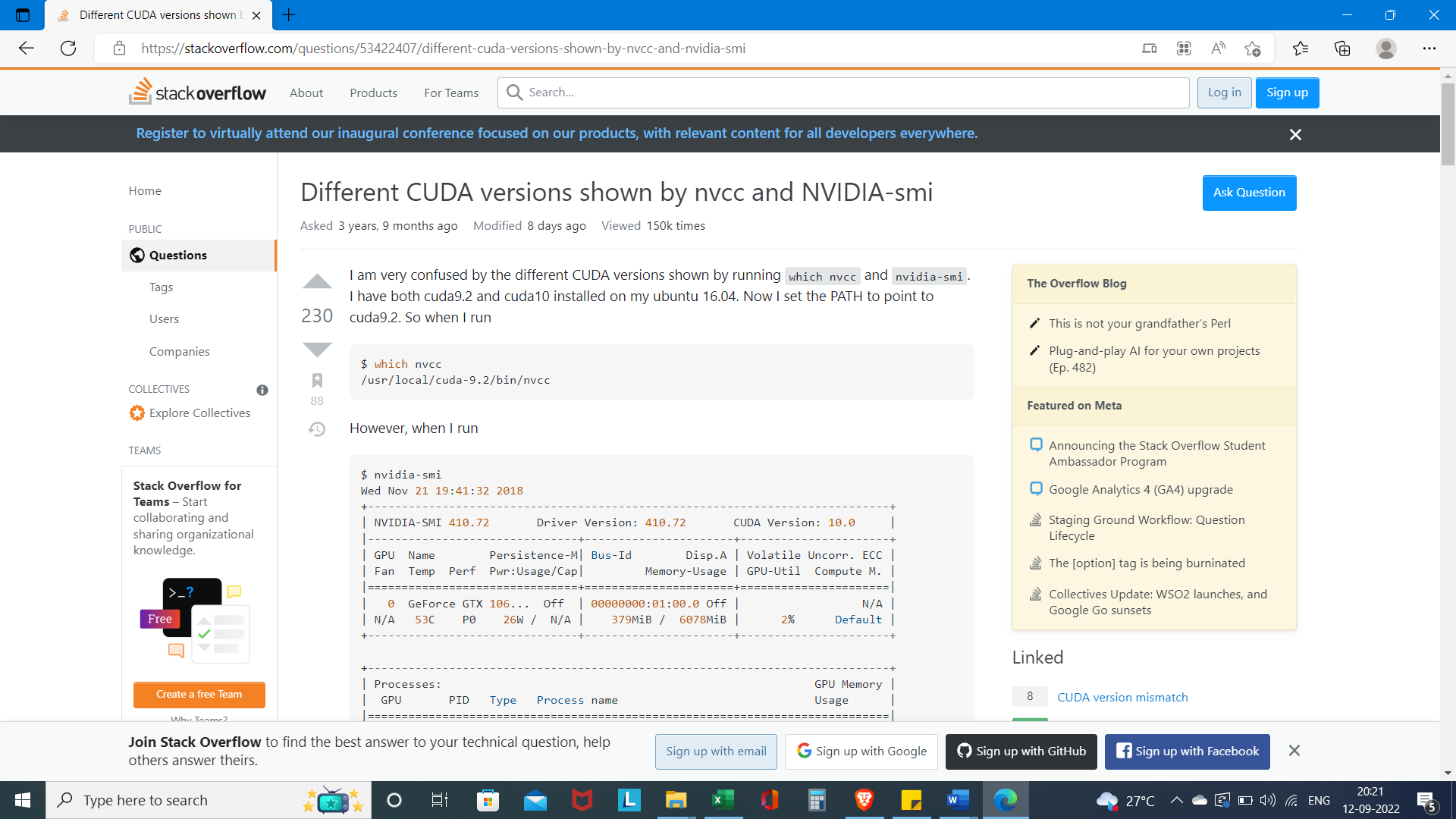
Description automatically generated

1. <https://stackoverflow.com/questions/53422407>

Graphical user interface, text, application, email

Description automatically generated

1. <https://stackoverflow.com/questions/53422407>



**Table I: Sample Data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Id** | 233446 | 36316243 | 1154174 | 1489757 | 2005331 |
| **PostTypeId** | 1 | 1 | 1 | 1 | 1 |
| **AcceptedAnswerId** | 233456 |  |  | 1492074 | 2005354 |
| **ParentId** |  |  |  |  |  |
| **CreationDate** | ######## | ######## | ######## | ######## | ######## |
| **DeletionDate** | |  |  |  |  |
| **Score** | 6 | 0 | 1 | 5 | 4 |
| **ViewCount** | 9894 | 500 | 737 | 95 | 2625 |
| **Body** | <p>I'm trying to write a small app that monitors how much power is left in a notebook battery and I'd like to know which Win32 function I could use to accomplish that.</p> | <p>I have a problem with the BeagleBoneBlack Rev A5C with Debian. I used this image: <code>BBB-eMMC-flasher-debian-8.3-lxqt-2gb-armhf-2016-03-20-2gb.img</code>. Debian 8.3 with Lxqt.</p>  <p>I want to develop a device for displaying a web page on full screen mode on startup and to be there for ever (the page autorefresh with some information).</p>  <p>Until this point, everything it's alright. I installed Midori, put it in the autostart with all the options for kiosk mode and it worked.</p>  <p>BUT, after sometime, the system goes black. When I move the mouse, it show the login screen. I need to disable this, I don't want to go to this screen never.</p>  <p>I have tried many things (installing XScreensaver, installing lxqt-powermanegement, touching the xset, etc.) and nothing seems to work.</p>  <p>I apologise for the bad english, not a native speaker.</p>  <p>Any help will be useful.</p> | <p>do the mobile phones have an API function, or something similar, that could be called by a phone application to determine whether the phone is currently charging or is on battery?</p>  <p>I'm guessing that if there is such a function, it would likely have a different name on each mobile OS, so if anybody could name those function names, and the OS for each, that would be great.</p>  <p>many thx</p> | <p>I'd like to be able to determine, programmatically, why a Vista system came out of standby (S3). Any language is fine, though a .NET one would be best. I'm looking to put this code into an application that I'm working on.</p>  <p>I'm looking for reasons like "Media center wanted to record a program," "user input woke the machine," "windows update is scheduled to run," things like that. I searched on Google, Bing and MSDN, but I could only find things relevant to Windows Mobile. </p>  <p>Can anyone point me to a function or object that can provide this information?</p> | <p>I would like create a service to do something when some "hot situation" occurs, and by "hot situation" I mean the following:</p>  <ul> <li>the GPS/cell coordinates are in a known zone</li> <li>a known Bluetooth device is detected</li> <li>a known Wi-Fi network is detected</li> <li>weather info has changed</li> <li>considerable change in acceleration of device detected: eg. from walking to travel by car.</li> </ul>  <p>As one can see, these tasks are "heavy" and will be run frequently â€“ say, 15 minutes, but sometimes even more â€“ when I have to detect the acceleration of the device.</p>  <p>The question I have is this: will this service drain the battery heavily? </p>  <p>If so, what recommendations do you have to minimize battery drain from the tasks described above?</p> |
| **OwnerUserId** | 9458 | 6136142 |  |  | 243782 |
| **OwnerDisplayName** | |  | mediterrano | user90784 | |
| **LastEditorUserId** | 418613 |  | 418613 |  | 418613 |
| **LastEditorDisplayName** | | |  | user90784 | |
| **LastEditDate** | ######## |  | ######## | ######## | ######## |
| **LastActivityDate** | ######## | ######## | ######## | ######## | ######## |
| **Title** | Monitor battery charge with Win32 API | BeagleBone Black as kiosk go to Screen Lock | How to programmatically check whether the mobile phone is currently charging or is on battery? | How can I determine why the machine came out of suspend? | how to minimize battery drain in a heavy service |
| **Tags** | <c++><c><winapi><monitor><power-management> | <debian><beagleboneblack><power-management> | <api><mobile-phones><power-management> | <windows-vista><power-management><standby> | <android><power-management> |
| **AnswerCount** | 4 | 1 | 2 | 1 | 2 |
| **CommentCount** | 0 | 0 | 0 | 2 | 0 |
| **FavoriteCount** | 1 |  |  | 2 | 2 |
| **ClosedDate** | |  |  |  |  |
| **CommunityOwnedDate** | | |  |  |  |
| **ContentLicense** | CC BY-SA 2.5 | CC BY-SA 3.0 | CC BY-SA 3.0 | CC BY-SA 2.5 | CC BY-SA 3.0 |

**Legend**

Id – Post ID

PostTypeId – Type of Post (Question, Answer, Wiki, TagWiki)

AcceptedAnswerId – Answers to the Question

ParentId – Answers are linked to questions via their parent id

CreationDate – Post created on

DeletionDate – Post deleted on

Score - number of upvotes on a question/answer minus the number of downvotes.

ViewCount – Number of Views

Body – Content of the Post

OwnerUserId - User ID of the Asker

OwnerDisplayName – User profile name displayed

LastEditorUserId – User ID of last editor

LastEditorDisplayName - Editor profile name displayed

LastEditDate – Date of last edit

LastActivityDate – Date of last viewed/edit/answer

Title – Title of Post

Tags – Tags used in the Post

AnswerCount – Number of Answers

CommentCount – Number of Comments

FavoriteCount – Number of likes

ClosedDate – Date if the questions is no longer active

CommunityOwnedDate – Stackoverflow generated date

ContentLicense – Stackoverflow license number for the Post

1. https://stackoverflow.com/questions/31326015 [↑](#footnote-ref-1)
2. https://stackoverflow.com/questions/50622525 [↑](#footnote-ref-2)
3. https://stackoverflow.com/questions/53422407 [↑](#footnote-ref-3)
4. https://stackoverflow.com/questions/25185405 [↑](#footnote-ref-4)
5. https://stackoverflow.com/questions/39649102 [↑](#footnote-ref-5)
6. https://stackoverflow.com/questions/25185405 [↑](#footnote-ref-6)
7. https://stackoverflow.com/questions/37063267 [↑](#footnote-ref-7)
8. https://stackoverflow.com/questions/31326015 [↑](#footnote-ref-8)
9. https://stackoverflow.com/questions/37319970 [↑](#footnote-ref-9)
10. https://stackoverflow.com/questions/39487553 [↑](#footnote-ref-10)
11. https://stackoverflow.com/questions/21826693 [↑](#footnote-ref-11)