Scalable Computing:
Practice and Experience

Scientific International Journal
for Parallel and Distributed Computing

ISSN: 1895-1767

Universitatea de Vest din Timişoara

Volume 23(1) March 2022

SUBSCRIPTION INFORMATION: please visit `http://www.scpe.org`

Scalable Computing: Practice and Experience

Volume 23, Number 1, March 2022

TABLE OF CONTENTS


Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2022 SCPE. Volume 23, Issues 1, pp. 1–8, DOI 10.12694/scpe.v23i1.1954

CUDA IMPLEMENTATION FOR EYE LOCATION ON INFRARED IMAGES

SORIN VALCAN* AND MIHAIL GAIANU†

Abstract. Parallel programming using GPUs is a modern solution to reduce computation time for large tasks. This is done by dividing algorithms in smaller parts which can be executed simultaneously. CUDA has many practical applications especially in video processing, medical imaging and machine learning. This paper presents how parallel implementations can speedup a ground truth data generation algorithm for eye location on infrared driver recordings which is executed on a database with more than 2 million frames. Computation time is much shorter compared to a sequential CPU implementation which makes it feasible to run it multiple times if updates are required and even use it in real-time applications.

Key words: CUDA; parallel programming; infrared camera; driver monitoring; eye detection

AMS subject classifications. 15A15, 15A09, 15A23

1. Introduction. Parallel processing is a modern way of accelerating the computation time for algorithms that can be divided into smaller parts, each of which is executed approximately simultaneously. This type of programming is most efficient when it is implemented on GPUs due to their special architecture that allows very fast execution of multiple threads.

This paper is focusing on programming using CUDA API provided by Nvidia for their GPUs. There are several CUDA applications that help researchers improve their computation time in areas such as bioinformatics, video processing, climate analysis, physics, gaming, machine learning and more. Time improvements are significant depending on the specific algorithm and how much it can be parallelized.

There are different forms of parallel computing such as instruction level, data parallelism or task parallelism. In GPU programming emphasis is on data parallelism because different sections of data can be processed in parallel using their multithreading capability.

This paper presents the use of CUDA in a ground truth data generator algorithm for eye location on infrared driver recordings which is presented in [1]. The purpose of this algorithm is to automatically generate good quality ground truth data that will be used for training neural networks in a system that will not require any human manual effort for data labeling.

In recent years, eye detection has become an important research topic in computer vision and pattern recognition ([3] [4]) , because the human eyes locations are essential information for many applications, including military, border control, facial expression recognition, auxiliary driving, and medical diagnosis [5]. For example, half of the face was covered in a cover test for detecting squint eyes [6].

Traditional eye detectors are typically designed according to the geometric characteristics of the eye. These eye detectors can be divided into two subclasses. The first subclass is the geometric model. Valenti and Gevers [7] used the curvature of isophotes to design a voting system for eye and pupil localization. Markuš et al. [8] proposed a method for eye pupil localization based on an ensemble of randomized regression trees. Timm and Barth [9] proposed the use of image gradients and squared dot products to detect the pupils. The second subclass is the template matching. The RANSAC [10] method was used to create an elliptic equation to fit the pupil center.

*Department of Computer Science, West University of Timişoara, 300223, Timişoara, Romania; Continental Automotive Romania VNI HMI, 300704, Timişoara, Romania(sorin.valcan96@e-uvt.ro).

†Department of Computer Science, West University of Timişoara, 300223, Timişoara, Romania; Continental Automotive Romania VNI HMI, 300704, Timişoara, Romania(mihail.gaianu@e-uvt.ro).

Multiple functionalities of our eye detection algorithm can be implemented using GPU programming and data parallelism concept. This has a big impact on the computation time required to generate ground truth data for recordings in our database and use them for training. It also makes it feasible to make updates for the algorithm and reprocess recordings in order to have better ground truth data without waiting weeks before starting the training process again.

The novelty of the paper is given by the remarkable capacity to reduce the time required for such an algorithm which is made up of multiple specific steps in order to have precise ground truth data.

2. Methods. This section describes the functionalities that were implemented in parallel using CUDA API and the focus of the discussion is on how it reduces the computational time required for eye detection.

This algorithm is made up of multiple steps because the resulted data must be very precise in order to train a neural network. By bringing a data parallelism approach in specific functionalities for each step the time is reduced significantly.

The main approach is based on the fact that pixels of a two dimensional image can be processed individually and simultaneously for various functionalities which is much faster than a sequential approach where each step(pixel) waits for the previous one to be done. Using this technique the functionalities discussed in this section are processed much faster.

2.1. CUDA API. CUDA API provides a general purpose programming model for the NVIDIA GPUs which helps the optimization and acceleration of algorithms that can be split in smaller tasks each of which can be executed in parallel.

2.1.1. Threads in CUDA. From a software point of view parallel programming is about finding a logical way of executing an algorithm in different threads in order to make it run faster. The CUDA API organizes threads in grids and thread blocks as described in [2].

For our work we used only one dimensional grids with the number of thread blocks computed using the formula:

$$B = \frac{T}{512} + 1 \tag{2.1}$$

where B represents the number of blocks, T is the number of threads required for the current functionality that is executed and 512 is a constant used for the number of threads to be contained within one block. Most of the modern GPUs have a maximum number of threads per block of 512 or 1028.

From a hardware point of view we are interested in understanding how the number of Streaming Multiprocessors (SM) and the warp size influences the way we should organize our algorithms.

A GPU has a specific number of SM and RAM memory which determines its performance. RAM memory is just the amount that can be stored in the GPU memory at a time. A SM is a unit responsible for executing blocks of threads. Blocks are distributed to SM as described in [2].

The warp size is the number of threads from a block that a SM can execute simultaneously from a hardware point of view and its value is 32. The access to threads waiting to be executed is done very fast which makes the entire process very efficient. This is the reason why it is very important to have the number of threads in a block multiple of 32, otherwise for each block there will be a final warp execution where some threads are not used which is a waste of parallel computing power for the entire grid execution.

2.1.2. GPU hardware specifications. For all computation times presented in our paper we used one GeForce RTX 2070 SUPER. This version has 40 multiprocessors which makes it very efficient because it can process up to 40 thread blocks in parallel. It also has 8 GB of RAM memory which allows it to have a big amount of data available on the GPU at a time but it does not have an influence on our algorithm performance because we use a small amount of memory.

2.2. Image representation in memory. Our infrared driver recordings contain 2D grayscale images with resolution 1280x800. We use the concept of flatten array to process each pixel individually on GPU because it is much easier to execute one memory allocation of the entire pixel data instead of a separate allocation for each line.

Once we have the flatten array to work with, in case the original indexes of the 2D image are needed we can use the following formulas to compute them:

$$line = \frac{idx}{w} \tag{2.2}$$

$$col = idx - (line * w) \tag{2.3}$$

where idx is the current index in the flatten array and w represents the 2D picture width.

2.3. Time comparison definition. In the following sections we provide for each functionality a time comparison between the sequential CPU and parallel GPU implementation. Usually when GPU time is discussed we need to take in consideration all steps described in Figure 2.1.

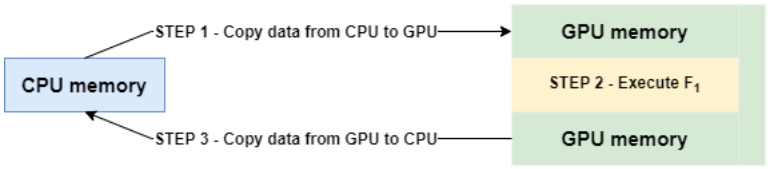

Fig. 2.1: Steps needed for executing single functionality on GPU

To exemplify this case we will consider a default time for data transfer between memories of 150 microseconds and for three GPU functions F_1, F_2, F_3 a time of 25, 30 and 35 microseconds respectively.

If for each individual function we will perform the three steps from above, we will have a complete execution time of 990 microseconds because the data transfer from CPU to GPU memory and back is executed three times

That is not the case in this paper because our entire algorithm is based on the structure defined in Figure 2.2.

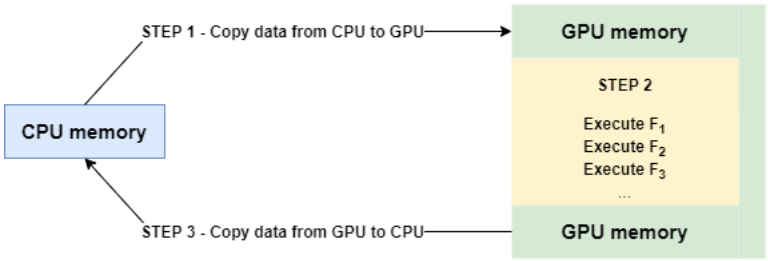

Fig. 2.2: Steps needed for executing multiple consecutive functionalities on GPU

When using this structure the time required to copy data between the two RAM memories has a much smaller impact on the total time required by the algorithm because data transfer must be performed only once(from CPU to GPU memory and back). Considering the same transfer and execution times from the previous example and running the GPU functions using the new sequence we will obtain a total execution time of 390 microseconds.

It is clear that the transfer time may vary depending on the size of the data being copied but the fact that we do not need to move data between the two memories multiple times makes the time improvement of each individual function F_n to be the main focus of this paper. This method of multiple function execution between two memory movements is a very efficient way of eliminating the big impact of data transfer time in GPU programming. It is understood that the design of the algorithm must allow such implementation which

may not always be possible for various technical reasons. The opposite statement is also true that algorithms design should be thought in a way that allows the use of such an efficient implementation.

The functionalities presented in the following sections represent an individual function F_n contained within a bigger algorithm $\{F_1, ... F_n ..., F_m\}$ which makes the transfer time a different discussion subject. For each functionality we will provide the average time needed on 100 different frames.

2.4. Picture conversion using threshold. The entire algorithm is based on grayscale images obtained from the infrared sensor which are converted to black-gray-white using two thresholds which are dynamically computed for each separate frame or eye patch. This conversion is implemented using the function described in Algorithm 1 which is executed in parallel using one separate thread on GPU for each pixel.

Algorithm 1 Conversion from grayscale to black-gray-white

```
Require: p                                          ▷ Flatten array of pixel values
  h, w                                              ▷ Image height and width
  al, ar, au, ad                                    ▷ Area limits where thresholds must be applied(left, right, up, down)
  t1, t2                                            ▷ Two thresholds for black-gray-white intervals
  procedure ApplyDoubleThreshold
    index ← blockDim.x × blockIdx.x + threadIdx.x   ▷ Method to compute index of current thread in the flattened array
    if index < w × h then
      line ← index / w
      col ← index − (line × w)
      if al ≤ col ≤ ar and au ≤ line ≤ ad then
        if p[index] < t1 then
          p[index] ← BLACK_PIXEL
        else if t1 ≤ p[index] ≤ t2 then
          p[index] ← GRAY_PIXEL
        else
          p[index] ← WHITE_PIXEL
        end if
      else
        p[index] ← BLACK_PIXEL                      ▷ If not in area, pixel becomes black
      end if
    end if
  end procedure
```

For our face area selection algorithm we use this functionality with the following boundary parameters:

- width: 1280
- height: 800
- up limit: 0
- down limit: 799
- left limit: 384
- right limit: 896
- t1 and t2 are dynamically computed for each frame in a separate function

This means there are 1,024,000 pixels to be processed. The time required for the sequential CPU implementation is 2.753 microseconds while the parallel GPU implementation requires 31 microseconds which makes it approximately 88 times faster.

2.5. Noise removal. This functionality is used to change the color of specific pixels in black-gray-white images according to the input parameters. One pixel will be marked with change flag if it is contained within one vertical and/or horizontal line with initial color that has pixels count less than an input maximum size. It is used in multiple algorithms like face area selection and eye selection for various logical reasons. The behaviour of the parallel implementation is described in Algorithm 2.

Algorithm 2 Mark noise pixels to be changed

```
Require: p                                  ▷ Flatten array of black-gray-white pixel values
  h, w                                      ▷ Image height and width
  rmPx                                      ▷ Flatten array of flags to mark color change needed
  rmSize                                    ▷ Maximum line size to be considered noise
  vert, horiz                               ▷ Boolean flags to search for noise on vertical, horizontal or both
  searchColor                               ▷ Color that will be replaced
  al, ar, au, ad                            ▷ Area limits where line dimensions are computed(left, right, up, down)
  procedure MarkPixelsToBeReplaced
    index ← blockDim.x × blockIdx.x + threadIdx.x   ▷ Method to compute index of current thread in the flattened array
    if index < w × h then
      line ← index / w
      col ← index − (line × w)
      if al ≤ col ≤ ar and au ≤ line ≤ ad then
        if p[index] = searchColor then
          if vert = true then
            lineSize ← VerticalCount()          ▷ Search up/down while pixels have searchColor
            if lineSize ≠ 0 and lineSize <= rmSize then
              MarkVerticalLineForReplacement(rmPx)   ▷ Same as VerticalCount() but it fills rmPx with replace flag
            end if
          end if
          if horiz = true then
            lineSize ← HorizontalCount()
            if lineSize ≠ 0 and lineSize <= rmSize then
              MarkHorizontalLineForReplacement(rmPx)
            end if
          end if
        end if
      end if
    end if
  end procedure
```

When this functionality is used in face area selection algorithm it has to mark 1,024,000 pixels but only between the limits described in Section 2.4. When it is executed using a sequential CPU implementation it requires 144.588 microseconds while the parallel GPU implementation requires 996 microseconds which makes it approximately 149 times faster.

When it is used in the eye selection algorithm it has to mark pixels on possible eye patches with resolution 70x70 which means 4.900 pixels. When it is executed using a sequential CPU implementation it requires 540 microseconds while the parallel GPU implementation requires 27 microseconds which makes it approximately 20 times faster.

2.6. Ratio map computation for eye patches. The ratio map is one of the scores we use to check if a possible eye patch actually contains an eye. This functionality marks each pixel with white in case it is contained within a horizontal and a vertical black line with ratio greater than 1.3. The advanced details of the ratio map score are presented in Section 2.4.1 from [1].

We implemented this functionality in parallel where a different thread uses a different pixel from the possible eye patch as a staring point to compute the ratio score. It is described in Algorithm 3.

We use this functionality only with the following parameters:

- width: 70
- height: 70

Algorithm 3 Compute ratio map for one possible eye patch

Require: p ▷ Flatten array of black-gray-white pixel values
h, w ▷ Image height and width
$outMap$ ▷ Flatten array of marked pixels
$searchColor$ ▷ Color for area of interest
$minRatio$ ▷ Minimum ratio between horizontal and vertical
al, ar, au, ad ▷ Area limits where line dimensions are computed(left, right, up, down)
procedure ComputeRatioMap
$index \leftarrow blockDim.x \times blockIdx.x + threadIdx.x$ ▷ Method to compute index of current thread in the flattened array
if $index < w \times h$ **then**
$outMap[index] \leftarrow BLACK\_PIXEL$
$line \leftarrow \frac{index}{w}$
$col \leftarrow index - (line \times w)$
if $al \leq col \leq ar$ and $au \leq line \leq ad$ **then**
if $p[index] = searchColor$ **then**
$verticalSize \leftarrow VerticalCount()$ ▷ Search up/down while pixels have searchColor
$horizontalSize \leftarrow HorizontalCount()$ ▷ Search left/right while pixels have searchColor
if $verticalSize \neq 0$ and $horizontalSize \neq 0$ **then**
$ratio \leftarrow \frac{horizontalSize}{verticalSize}$
if $ratio \geq minRatio$ **then**
$outMap[index] \leftarrow WHITE\_PIXEL$
end if
end if
end if
end if
end if
end procedure

- searchColor: BLACK_PIXEL
- minRatio: 1.3
- down limit: 69
- up limit: 0
- left limit: 0
- right limit: 69

When this functionality is executed using a sequential CPU implementation it requires 452 microseconds while the parallel GPU implementation requires only 33 microseconds which makes it approximately 13 times faster.

3. Obtained Results. Until now we presented time improvements obtained for separate functionalities by implementing them in parallel using GPU and CUDA API. In this section we will present an overall time improvement for our ground truth data generator.

All these functionalities that we presented are individual functions which work more efficient in a parallel implementation compared to a sequential CPU version. They are all part of a big eye detection algorithm presented in [1].

For our overall time improvement computation we ran a version of the algorithm where only the three functionalities presented above are implemented in a sequential CPU way. The rest of the algorithm may still contain GPU implementations which are more efficient. In Table 3.1 we present the time improvements obtained only with the functionalities described in this paper.

We selected 10 random recordings with different number of frames and different number of ground truth data frames computed. We can see on the time improvement column that on average the GPU parallel implementation is 15.88 times faster than the sequential CPU implementation. For recordings where a bigger

Table 3.1: Time improvements for GPU implementations compared to sequential CPU

Recording name	Number of frames	Ground truth frames generated	CPU time(s)	GPU time(s)	X times faster	GPU average fps
Rec 1	5422	363	2418	162	14.92	33.46
Rec 2	3194	674	1433	95	15.08	33.62
Rec 3	1421	327	599	35	17.11	40.60
Rec 4	3103	2047	1429	80	17.86	38.78
Rec 5	5879	751	2509	173	14.50	33.98
Rec 6	1995	415	965	59	16.35	33.81
Rec 7	5934	321	2726	171	15.94	34.70
Rec 8	885	70	378	24	15.75	36.87
Rec 9	6320	714	2724	178	15.30	35.50
Rec 10	1737	609	736	46	16.00	37.76

Fig. 3.1: Example of eye labels ground truth data generated for one frame

number of ground truth frames is generated like Rec 4 we can see the best improvements because the algorithm is searching in a smaller area when the eye locations for the previous frame is available which makes it more efficient.

This results have huge implications when we want to generate ground truth data on 2 millions frames (see example of eye labels in Figure 3.1). Just by taking the worst case of Rec 5 we can estimate the time required on GPU would take less than 17 hours compared to the CPU version of the algorithm which would take almost 10 days. This makes the entire process of ground truth data generation to be more feasible and more prone to improvements because it can be ran multiple times in a short period of time.

Some recordings have better improvements compared to others because there are less frames with face area detected. This leaves no room for time improvements in the eye detection functionalities because they never get to be executed. We can observe in cases like Rec 1 and Rec 5 that a small number of ground truth frames are generated which leads to a smaller time improvement. There are also cases like Rec 7 where we have a small number of ground truth frames but the time improvement is better. It means that face area was detected, the eye detection algorithm was executed but eyes were not found very often.

The GPU average fps column is computed by diving the number of frames to the GPU time required to process the recording. We can observe the minimum fps is 33 which makes the algorithm feasible to be run in real-time scenarios. This was not the main purpose of our parallel implementations but the performance one can obtain using parallel programming is remarkable.

4. Future Work. In future publications we will present GPU implementations we are using in the processing of possible eye patches. This requires the generation of the possible eye patches for a given area in a parallel way and storing it in a single continuous location of memory on the GPU. Once the eye patches are available we can use this memory location to perform steps for eye selection by processing each possible patch in a separate thread where this is possible.

In the future development of the ground truth data generator we want to include nostrils and mouth detection in order to have all face features available for training. Those algorithms will use similar parallel functionalities and we expect the time improvement of the entire detection(eyes, nostrils, mouth) to become even more significant compared to sequential CPU implementations.

5. Conclusions. This paper presented three parallel implementations which are used for eye detection on infrared driver recordings. The time improvement is significant and has a big impact on the entire process of ground truth data generation on a big number of frames. It also makes the algorithm feasible to be updated and re-executed in a short amount of time and it can be also used for real-time detection due to good fps performance.

Such parallel implementation methods can be used in several areas to reduce the time required for specific computations and have a major impact for big data processing where efficient algorithms help us save weeks of waiting time.

REFERENCES

[1] Valcan, S. and Gaianu, M., *Ground Truth Data Generator for Eye Location on Infrared Driver Recordings*, in Journal of Imaging 7, 162, doi: 10.3390/jimaging7090162, 2021.
[2] *CUDA C++ Programming Guide*, Available at https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html
[3] H. Fu, Y. Wei, F. Camastra, P. Arico, and H. Sheng, *Advances in Eye Tracking Technology: Theory, Algorithms, and Applicationst*, Computational Intelligence and Neuroscience, vol. 2016, Article ID 7831469, 2016.
[4] L. Zhang, Y. Cao, F. Yang, and Q. Zhao, *Machine Learning and Visual Computing*, Applied Computational Intelligence and Soft Computing, vol. 2017, Article ID 7571043, 2017.
[5] H. Mosa, M. Ali, and K. Kyamakya, LU-*A computerized method to diagnose strabismus based on a novel method for pupil segmentation*, in Proceedings of the International Symposium on Theoretical Electrical Engineering, 2013.
[6] Lorenz, Birgit and Moore, Anthony. (eds.) *Pediatric Ophthalmology, Neuro-Ophthalmology, Genetics*, doi: 10.1007/3-540-31220-X, 2006.
[7] R. Valenti and T. Gevers, *Accurate eye center location through invariant isocentric patterns*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 9, pp. 1785–1798, 2012.
[8] N. Markuš, M. Frljak, I. S. Pandžić, J. Ahlberg, and R. Forchheimer, *Eye pupil localization with an ensemble of randomized trees*, Pattern Recognition, vol. 47, no. 2, pp. 578–587, 2014.
[9] F. Timm and E. Barth, *Accurate eye centre localisation by means of gradients*, in Proceedings of the 6th International Conference on Computer Vision Theory and Applications, VISAPP 11, pp. 125–130, 2011.
[10] L. Świrski, A. Bulling, and N. Dodgson, *Robust real-time pupil tracking in highly off-axis images*, in Proceedings of the 7th Eye Tracking Research and Applications Symposium, ETRA 2012, pp. 173–176, 2012.

Edited by: Marian Vajtersic
Received: Dec 17, 2021
Accepted: Apr 6, 2022

Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2022 SCPE. Volume 23, Issues 1, pp. 9–21, DOI 10.12694/scpe.v23i1.1957

EXPLORING USABILITY OF REDDIT IN DATA SCIENCE AND KNOWLEDGE PROCESSING

JAN SAWICKI*, MARIA GANZHA†, MARCIN PAPRZYCKI‡, AND AMELIA BĂDICĂ§

Abstract. This contribution argues that Reddit, as a massive, categorized, open-access dataset, is a useful data source, for "almost any topic". Hence, it can be used in data science, e.g. for knowledge exploration. This statement is backed-up with presented analysis, based on 180 manually annotated papers, related to Reddit itself, and data acquired from popular databases of scientific papers. Finally, an open source tool is introduced, which provides an easy access to Reddit resources, and an exploratory data analysis of how Reddit covers selected topics. These functions can be used as a prelude analysis to a broader exploration of Reddit's applicability.

Key words: Reddit, online forum, dataset, text mining, information retrieval, data analytics, knowledge processing

AMS subject classifications. 68-02, 68U15, 68U99, 68U01, 68T50, 91Fxx

1. Introduction. Recently, social networks and content sharing networks became popular repositories of data, used for information and knowledge processing (especially for information retrieval). The aim of this work is to explore the usability of Reddit as a data source. In this context, we present a review of scientific literature about Reddit itself, its presence in scientific databases, and elaborate its "topical coverage". Moreover, for the latter study, a specialized tool (*Reddit-TUDFE*)' is introduced, which allows for fast check of Reddit coverage of a selected topic. The key contributions of this work are answers to the following research question (RQs):

- **RQ1**: What are the most popular methods to acquire Reddit data? (do they allow capturing graph networks[1])
- **RQ2**: What problems are the most researched when using Reddit as a dataset?
- **RQ3**: How does Reddit usage in data science change over time? Is it declining or is it increasing?
- **RQ4**: Are there any popular topics that are not (substantially) covered on Reddit?
- **RQ5**: Is Reddit used as a single dataset, or with datasets from other online platforms?[2]

These questions are essential for further planned research and positive answer would mean that Reddit is a proper choice for proceeding with the project of information retrieval about popular trends, using graph databases and complex networks. Moreover, positive answers would indicate that Reddit may be a competitor (or a companion) to explorations based on more popular data sources, like Twitter.

2. What is Reddit. Let us start from a brief description of Reddit. It is a web content rating and discussion website [31]. It was created in 2005 and is ranked as the 17$^{\text{th}}$ most visited website in the world, with over 430 million monthly active users[3] and total of over 13 billion posts and comments[4]. The structure of Reddit is illustrated in Figure 2.1.

Reddit is divided into thematic subfora (so called, *subreddits*) dynamically created by its users. Therefore, the topic structure is systematically evolving, in response to user needs. Each subreddit has its *moderators*

*Warsaw University of Technology, Department of Mathematics and Information Sciences, (jan.sawicki2.dokt@pw.edu.pl).
†Warsaw University of Technology, Department of Mathematics and Information Sciences (m.ganzha@mini.pw.edu.pl).
‡Systems Research Institute Polish Academy of Sciences (marcin.paprzycki@ibspan.waw.pl).
§University of Craiova, Department of Statistics and Business Informatics (amelia.badica@edu.ucv.ro).

[1]Here, graph networks are of special interests, because it can be observed that large number of methods of data extraction and analysis are focused on application of graph theory.

[2]Answer to this question is crucial to establish (suggest) additional datasets, which could/should be used with Reddit.

[3]https://www.statista.com/topics/5672/reddit/#dossierSummary

[4]https://www.redditinc.com/

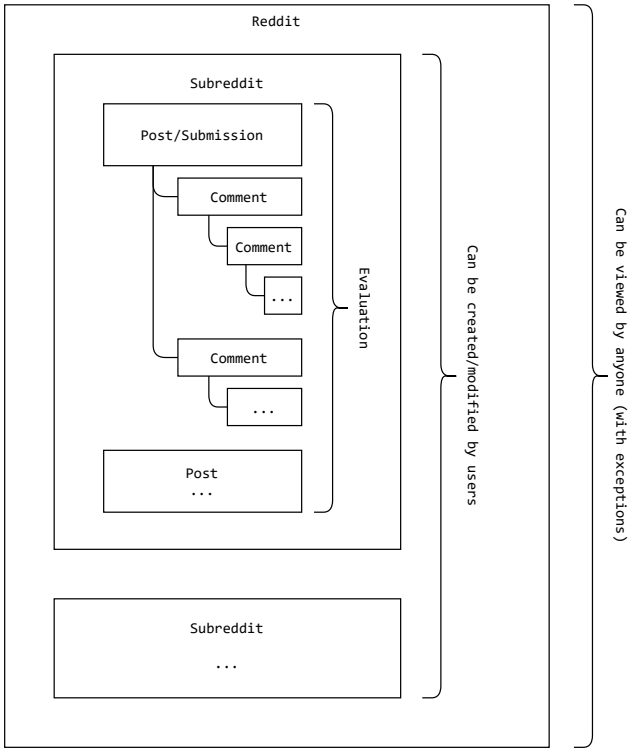

Fig. 2.1: Reddit structure

who may supervise *submissions* and *comments*. Comments are linked to submissions, or to earlier comments, forming a tree-like structure.

2.1. Content access rules and restrictions. Most of the subreddits are public (for registered and non-registred users). There are some exceptions based, for instance, on karma points (i.e. user's score), comments, gold (i.e. Reddit's currency that can be purchased with real money), moderator status, time on Reddit, username and others. For instance, such restriction can be be applied to even a Harry Potter house preference (e.g. r/gryffindor)[5]. Here, let us note that the Reddit topic explorations tool (introduced in Section 5), is based only on access to publicly available data.

2.2. Accessibility – Reddit API vs. Pushshift API. Not only is the data on Reddit publicly accessible (with the exception of private communities), it is also made available via the official Reddit API[6]. However, in the course of literature review, it was found that most researchers do not actually use it. Over 90% of analyzed papers either use ready datasets scraped earlier from Reddit and posted online (possibly in an annotated form), or they choose the Pushshift API [4]. None of the analysed papers stated the explicit reason for this choice (very few even mention how their datasets have been retrieved). However, practically testing capabilities of Reddit API and Pushshift API shows that the key factor could have been that Reddit API does not allow easy retrieval of historical data, while Pushshift API does. Hence, when developing the Reddit data exploration tool, the Pushshift API was used.

3. Data acquisition and processing. To explore Reddit, as seen by the scientists, a dataset of all, most recent, papers available on arXiv has been assembled – a total of 180 papers. All of them were related to Reddit

[5]`https://www.reddit.com/r/ListOfSubreddits/wiki/privates`
[6]https://www.reddit.com/dev/api/

and submitted to arXiv between 01-01-2019 and 01-03-2021 (and retrieved on 30-03-2021[7]). This dataset has been processed both manually and automatically. First, collected papers have been manually annotated with four attribute sets: **topic** (a general area of research), **methods** (theoretical approach, e.g. neural network, text embedding), **dataset** and **technologies** (practical software, e.g. BERT [10]). Next, obtained results were merged using arXiv identification code and the publicly available data, i.e. the content (title and raw text) and the bibliometric metadata. This allowed extraction of information presented in Section 4. All collected content has been converted to a raw text file, using PDF Miner software [38]. Next, the key features of titles and texts have been cleaned and mined using the NLTK framework [26] (for sentiment and subjectivity), and TF-IDF [36] for vectorization (both frameworks are part of the scikit-klearn library [34]).

4. Analysis and findings. As a result of processing of collected data, we were able to formulate a number of observations. Let us summarize the most important ones.

4.1. Metadata and bibliometrics. First, let us consider a few noticeable bibliometric and authorship statistics, gathered using Semantic Scholar[8] and presented in Figures 4.1, 4.2 and 4.3.

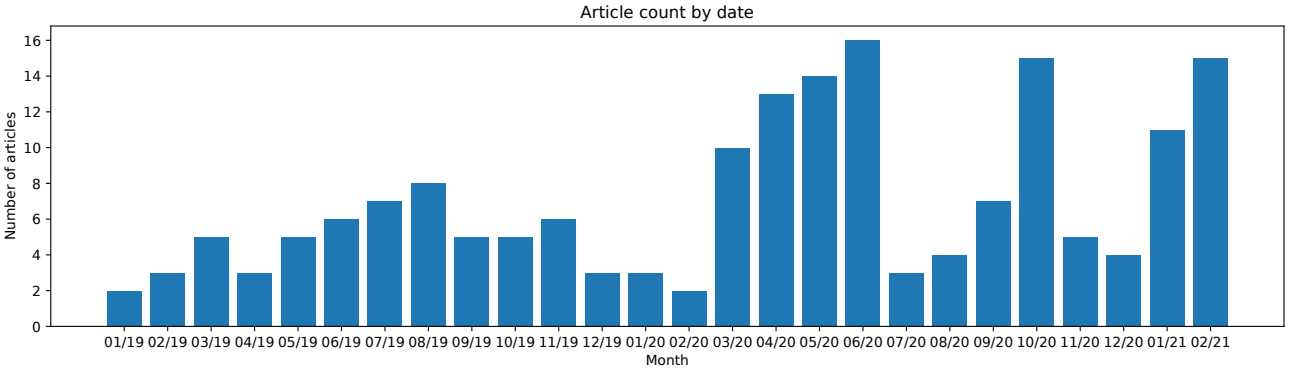

Fig. 4.1: Article count by the month of the submission date

As shown in Figure 4.1, there is a significant growth in the number of articles (related to Reddit) published after March 2020 (correlated with the outburst of the COVID-19 pandemic) and in October 2020 (correlated with notification dates for many scientific conferences [40]). The latter fact was also verified during manual processing of collected data. This suggests that Reddit was used to provide data related to COVID pandemic and that it is used as a data source for contributions to, broadly understood, data analytics related conferences.

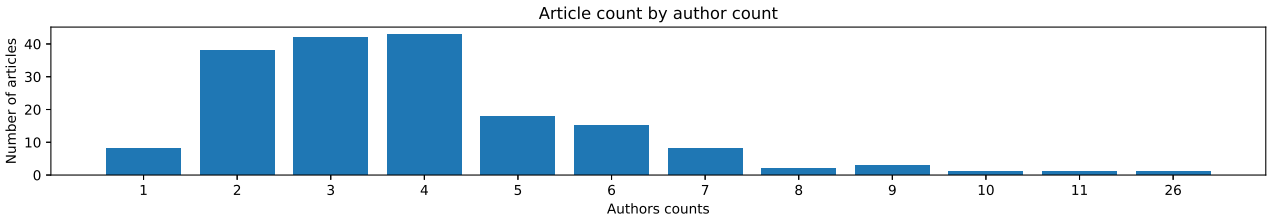

Fig. 4.2: Number of authors for the selected papers

Next, as seen in Figure 4.2, majority of papers were written by 2-4 authors, with one having 26 authors [13].

[7] https://arxiv.org/search/advanced?&terms-0-term=reddit&classification-computer_science=y&date-from_date=2019-01-01&date-to_date

[8] https://api.semanticscholar.org/

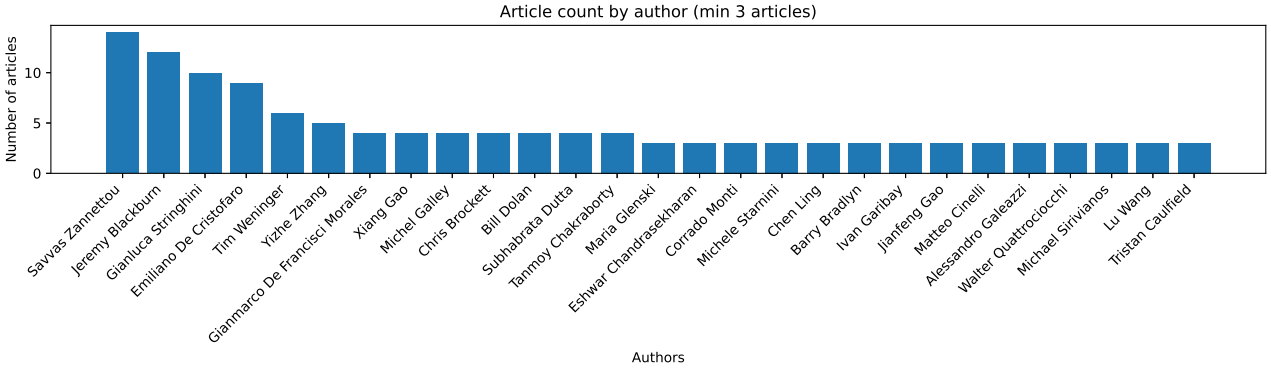

Fig. 4.3: Article count by the author

Finally, Figure 4.3, shows that the most prolific authors, of Reddit-related papers, were Savvas Zannettou (Max-Planck-Institute), Jeremy Blackburn (Binghamton University) and Gianluca Stringhini (Boston University). This seems to suggest that large number of scientific content, generated while studying Reddit posts, is delivered by a close circle of scientists.

4.2. Analysis of topic, methods and technology. Topics, methods and technologies are key to answer RQ1 and RQ2. These were extracted manually from the collected papers. They are summarized in Figures 4.4, 4.5 and 4.6.

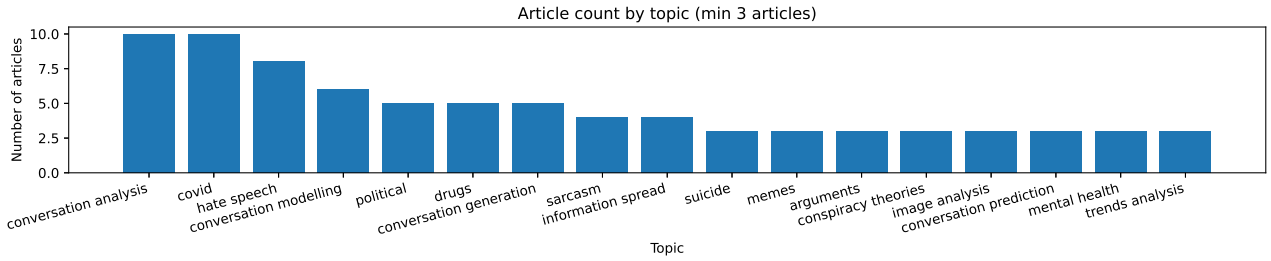

Fig. 4.4: Article count by article topics manually annotated in all papers

Figures 4.4 and 4.5 show clearly that the most popular research topic is *conversation*, which matches the fact that Reddit is a discussion forum. Due to the timing of this work (overlapping with the COVID-19 pandemic), the second most common topic is *COVID* (see Figure 4.4).

Since Reddit consists mostly of text-based discussions, it is not surprising that the two most common methods, in Reddit-related research, are *text embeddings*, used in text processing, and *networks*, used for social network analysis. Note that, in the reported results, "network" (understood as a graph) and "neural network" are separate terms.

Regarding technologies (shown in Figure 4.6), over 45% of studies used Pushshift API [4] for Reddit data extraction, and over 35% applied BERT [10] embedding (and its variations) for the natural language processing.

Finally, topics and methods have been combined in a correlation heatmap (Figure 4.7).

Here, a few significant correlations have been established. However, they have to be considered keeping in mind that they materialize in the context of a specific dataset, created on from contributions reporting research that used Reddit as a data source. therefore, no claim is made that these observation can be immediately generalized beyond the dataset used in this work. However, based on general knowledge of the field, they seem to be in line with more general trends.

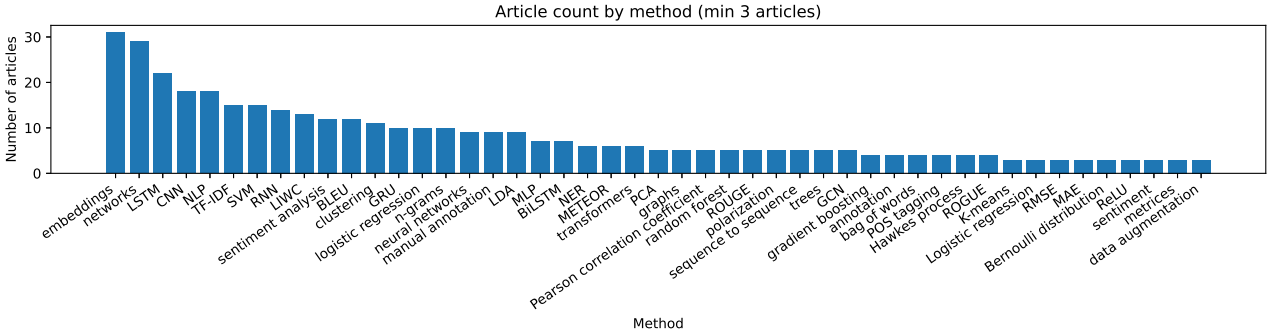

Fig. 4.5: Article count by methods manually annotated in all papers

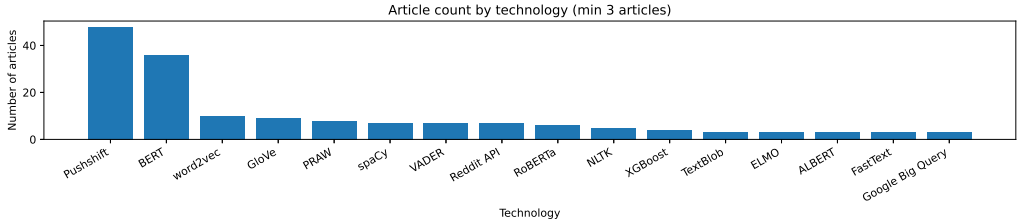

Fig. 4.6: Article count by technologies manually annotated in all papers

- Papers related to *drugs* typically use *word embeddings*. However, this can be related to the overall popularity of word embeddings in the research conducted in early 2020th (see, for instance, the citation count for [10]).
- *Networks* are typically applied in analysis of *trends*, e.g. topic popularity (this is a key finding for RQ1).
- Articles dealing with *sarcasm* often use *LSTM networks*.
- Research devoted to the *conversation generation* typically applies the *BLEU metric*.

4.2.1. Topics of knowledge and information processing. The topic of information and knowledge retrieval is one of the main aims of undertaken analysis. Hence, this category was checked specifically. Even though many works focus on information spreading in online communities [13, 41, 11, 12], there is hardly any focus purely on information/knowledge retrieval. There are precisely two papers (1% of the considered work) related to knowledge processing (specifically, knowledge graphs [6, 43]). Expanding arXiv search, to capture all articles including terms "knowledge" and "Reddit", resulted in 4 records, none of which is related to knowledge capture. Pairing keyword "Reddit" with "information retrieval" or 'information processing" yielded 0 results. Therefore, top knowledge processing/management-related conferences were searched, but only one contribution [18], about knowledge and Reddit, has been found (published by the K-CAP conference in 2011[9]). This renders Reddit as a source that is definitely underexplored in terms of knowledge/information mining.

4.2.2. Use of Reddit combined with other datasets. Moving to the **RQ5**, it was discovered that among papers that use Reddit, over 30% also use Twitter, which is a data source that is very often used for sentiment analysis [24]). Other datasets that have been utilized together with Reddit are: Facebook, 4Chan, YouTube, and Gab. Each of them appears in less than 10% of papers, which used Reddit (details are shown in Figure 4.8). Datasets are rarely used in triplets, i.e. Reddit and two other datasets (the highest scoring triplets were Reddit, combined with Twitter and Facebook 6.6% of articles; Reddit, used together with Twitter and

[9] `https://www.k-cap.org/kcap11/index.html`

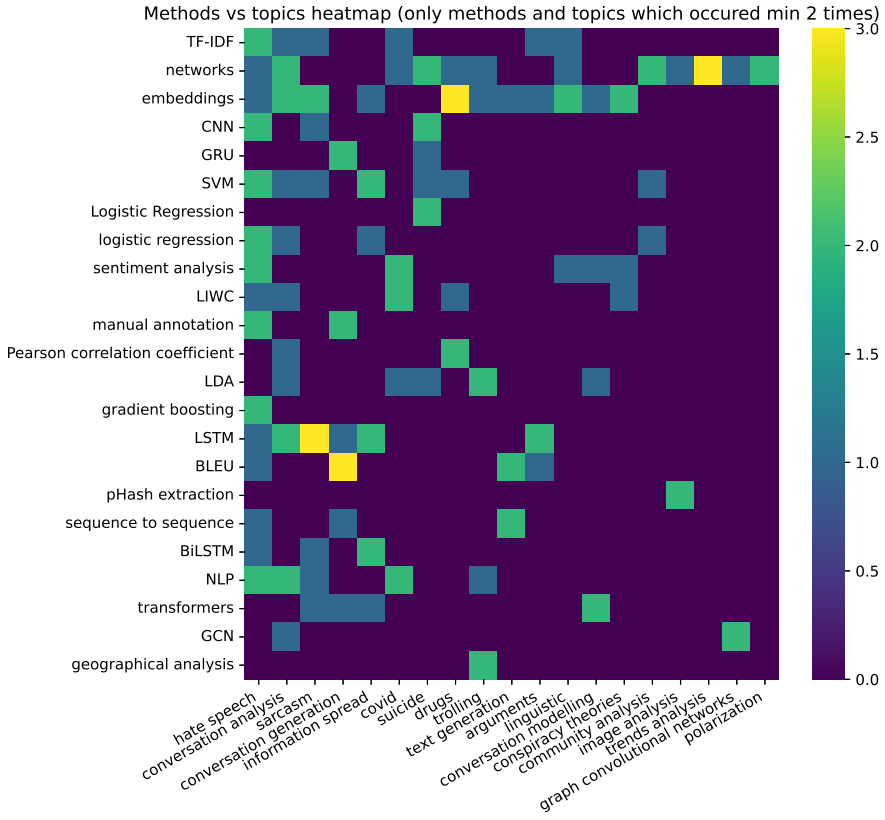

Fig. 4.7: Research methods correlated with article topics

4chan 6% (e.g. [41, 42]), and Reddit studied jointly with Twitter, YouTube 5% of contributions (e.g. [5])). Finally, a single paper considers combination of four datasets (i.e. Reddit, Twitter, Facebook, and Gab [7]).

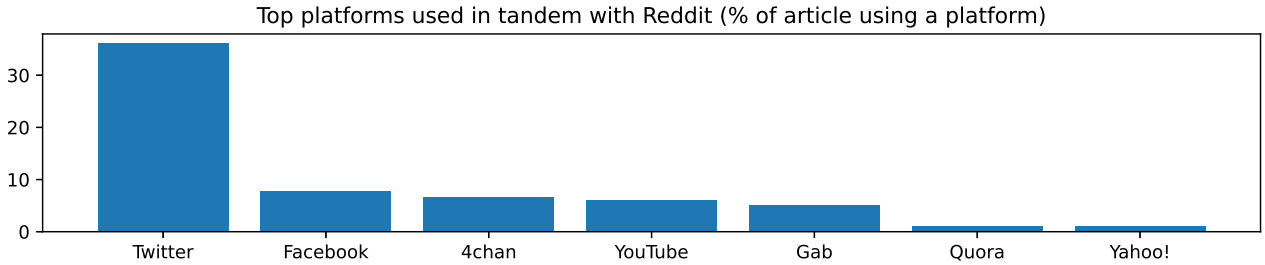

Fig. 4.8: Online platforms used as data sources together with Reddit

An interesting use case of Reddit usage in scientific environment has been found in "IEEE Top Programming Languages: Design, Methods, and Data Sources"[10]. This work shows a practical approach to a an interesting research question; here, what are the top programming languages. In this work Reddit is listed as one of the sources among others, such as Google Trends, Twitter, GitHub and Stack Overflow.

[10] `https://spectrum.ieee.org/ieee-top-programming-languages-design-methods-and-data-sources`

4.3. Linguistic analysis. During exploratory data analysis, various natural language processing techniques were applied. Among them, papers were also analysed linguistically. Specifically, sentiment analysis using NLTK framework [26] and SentimentAnalyzer[11] was applied. Observed polarization (depicted in Figure 4.9) indicates a negligible displacement towards the positive sentiment. This was expected, and is consistent with previous studies on scientific literature sentiment [19].

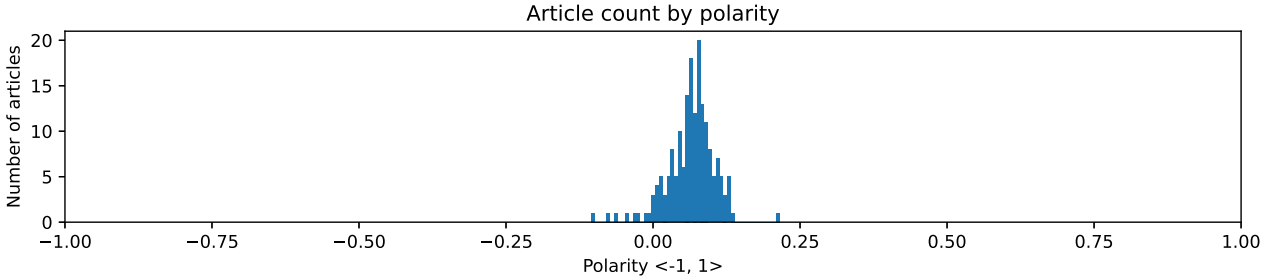

Fig. 4.9: Histogram of number articles based on text polarity measures

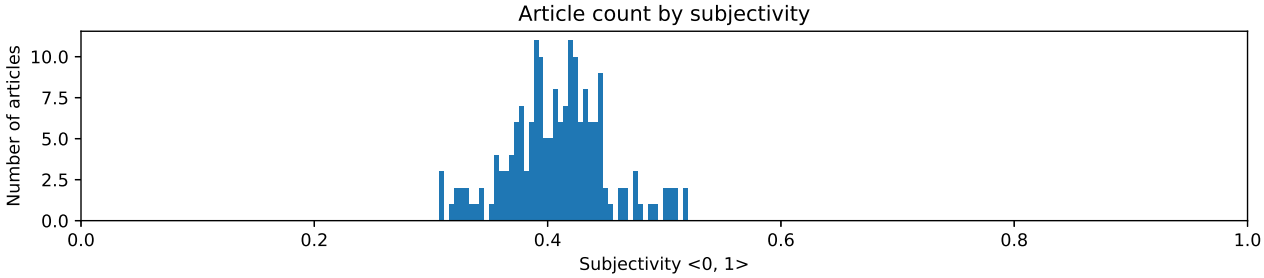

Fig. 4.10: Histogram of number articles based on text subjectivity measures

However, the subjectivity measure (summarised in Figure 4.10) raised concerns. Obviously, it has been claimed that scientific research may be subjective, as it needs to allow "leaps of faith" (see, [9]). Moreover, some philosophers [30, 29] argue that subjectivity is intrinsic for human nature. However, it is also claimed (and for good reasons) that the foundation of the scientific method [32] revolves around aiming at objectivity. Hence, results summarised in Figure 4.10, indicating high level of subjectivity, were somewhat concerning. To establish the reason for this finding, the most "subjective" texts were studied directly. As a result is was found that this is a false alarm. Specifically, apparent shift towards subjectivity was caused by inaccuracy of the classifier (*SentimentIntensityAnalyzer* from *nltk.sentiment*[12]). For further understanding, let us consider the selected sentences from the most subjective (according to the NLTK metric) articles.

- "However, this openness formed a platform for the polarization of opinions and controversial discussions" [22] (score: 0.95)
- "(...) also presented an extended version of the study discussing potential racial bias in offensive content datasets (...)" [2] (score: 1.0)
- "All datasets only contain activity between 01/2015 and 10/2018" [16] (score: 1.0)

Moreover, let us also consider how the calculated subjectivity measure changes with a simple modification of selected statements (i.e. by removing particular words):

[11] `https://www.nltk.org/api/nltk.sentiment.sentiment_analyzer.html`
[12] `https://www.nltk.org/api/nltk.sentiment.html`

- Statement before transformation (score: 0.63):
  "Controversially initiated and non-controversially initiated cascades, (a,b,c) are controversially initiated posts' cascades while (d,e,f) are non-controversial posts' cascades where the red dots represent a comment labeled as controversial by Reddit that is directed to the post's author while a green dot is a comment labeled controversial by Reddit that is directed to another comment." [22]
- The same statement after transformation (score: 0.15):
  "initiated and initiated cascades, (a,b,c) are initiated posts' cascades while (d,e,f) are posts' cascades where the red dots represent a comment labeled as by Reddit that is directed to the post's author while a green dot is a comment labeled by Reddit that is directed to another comment." [22]

This suggests that simply using the "subjective" (key)words (e.g. "controversial", "bias") in the text, regardless of their context, results in radically increased value of the variable that is to indicate subjectivity of the text. However, there are sentences that do not use such words, which have also received a high subjectivity score. Hence, further research would be required into the way that the NLTK metric works and why, sometimes, it is rather misleading. However, this is outside of scope of the current contribution.

4.4. Reddit-based literature in scholarly databases. Let us now address **RQ3** and **RQ4**. Even though they cannot be unequivocally answered, possible answers can be experimentally explored. To verify the change over time of the number of scholarly papers related to Reddit, between 2010 and 2021, 10 databases have been analysed and queried for the term "reddit". As shown in Figure 4.11 the number of found articles raises year to year (**RQ3**).

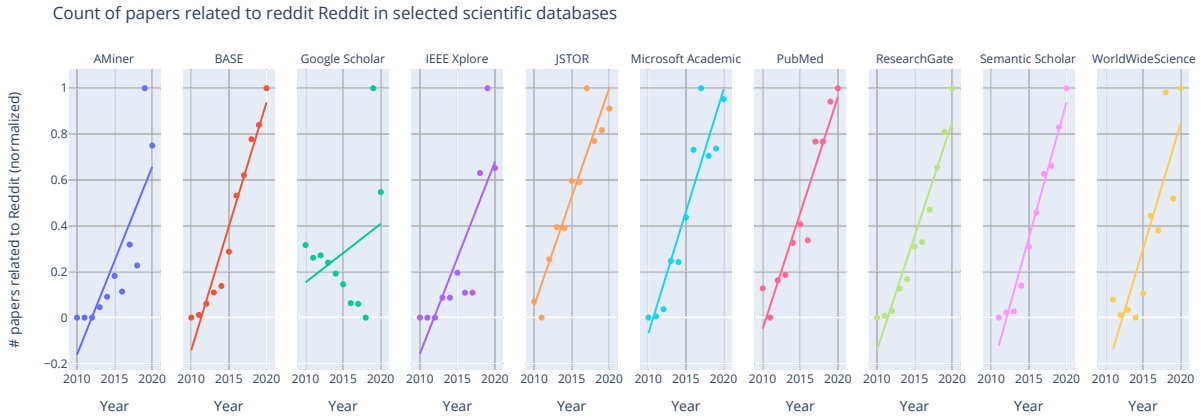

Fig. 4.11: Non-cumulative count of papers related to Reddit in scientific databases in years 2010-2021.

Table 4.1 shows how many articles, related to Reddit (e.g. using it as a data source, processing it, analysing the comments, etc.), have been indexed in scientific databases.

Observations that can be made, on the basis of the results found in Table 4.1, are:

- the number of Reddit articles is quite small, yet representative,
- the number of Reddit papers is somewhat proportional in each database, so it can be stated that the literature is quite equally spread in the Internet.

4.4.1. Outlying results found in Google Scholar. The only database with trends inconsistent with others is Google Scholar (Figure 4.11). However, although it is one of the most widely known databases that indexes scientific publications [27, 17], it already received both praise and criticism [20, 39]. Main problems of Google Scholar, pointed out in the literature, are: (i) difficulties to estimate the actual size of the database [33, 21], (2) gender- and race-related bias in displaying contributions [23], (iii) favoring incremental work [23], (iv) favoring larger research communities [23], (v) limited indexing of files [20], (vi) incorrect biblio-

Table 4.1: Number of scientific papers (which appeared in 2011-2021) related to Reddit (e.g. using it as dataset, exploring its structure etc.) indexed in selected databases (all accessed on 09-06-2021).

Database	Total size	# papers
Google Scholar	330M [15]	980[13]
JSTOR	12M[14]	2555[15]
PubMed	32M[16]	243[17]
AMiner	230M [15]	840[18]
Bielefeld Academic Search Engine	270M[19]	5176[20]
Semantic Scholar	197M[21]	2280[22]
AMiner	320M[23]	784[24]
Microsoft Academic	170M [15]	902[25]
WorldWideScience	300M [15]	1250[26]
IEEE Xplore	8551[27]	162[28]
ResearchGate	135M[29]	3001[30]

metrics (due to automated algorithms instead of skilled librarians) [28, 14], (vi) "uncertain quality of Google Scholar's performance" [14], (vii) "Google Scholar's inability or unwillingness to elaborate on what documents its system crawls" [14], and (viii) limitations of bibliometric analysis [28]. Moreover, Google Scholar declares inconsistently the number of results of a query, and the actual number of returned results (e.g. a query returns 1000 actual results, while it declares 58,600[31]). This finding may correspond to already reported Google Scholar inconsistencies [33, 21] and lack of transparency [14]. Therefore, Google Scholar can be treated as an outlier and disregarded in conclusions drawn from this experiment.

4.5. Google Trends. The next experiment explored presence of popular trends in Reddit. This was done based on Google Trends, an analytical website which provides information about popularity of search queries in Google search engine [32]. For all Global Google Trends 2020[33] their Reddit presence has been measured (see Table 4.2). Overall, 79% of top Google Trends have a dedicated subreddit, while *all of them* are widely discussed. Table 4.2 illustrates top three in each Google Trend category.

5. Reddit as "The Ultimate Dataset for Everything". To further study whether Reddit contains information about (almost) "any area", a tool for easy exploratory data analysis (EDA [8]) was designed. Specifically, *Reddit-TUDFE* allows quick search of any topic on Reddit, checking if/how it is represented, and how it is discussed. Specifically, *Reddit-TUDFE* delivers the following functions:

1. Uses Reddit API to search for best matching subreddit.
2. Downloads newest N posts from the subreddit, using Pushshift API and a combination of PRAW[35] and PSAW[36].
3. Performs basic text cleaning (tokenization with NLTK [26], removal of stopwords, punctuation, numbers).
4. Generates and displays post titles and content wordclouds[37].

The code follows state-of-the-art solutions for code sharing ([35]) and is publicly available on GitHub[38] as a Jupyter Notebook [37].

To illustrate the capabilities of the developed application, let us present few examples, in two groups, in

[31] https://scholar.google.com/scholar?start=990&q=reddit&hl=en&as_sdt=0,5&as_ylo=2020&as_yhi=2020 accessed on 11-09-2021

[32] https://trends.google.com/trends

[33] https://trends.google.com/trends/yis/2020/GLOBAL/

[35] https://github.com/praw-dev/praw

[36] https://github.com/dmarx/psaw

[37] https://github.com/amueller/word_cloud

[38] https://anonymous.4open.science/r/reddit-tudfe-B736/reddit_tudfe.ipynb

https://anonymous.4open.science/r/reddit-tudfe-B736/reddit_tudfe.ipynb

Table 4.2: Global Google Trends 2020[34] (top 3 in each Google Trends category) and their appearance on Reddit ("subreddit" – there exists a dedicated subforum, "discussion" – the topic is present in (a) subreddit(s) of a broader topic)

Google Trend	category	on Reddit	reference
Coronavirus	searches	subreddit	r/Coronavirus
Election results	searches	discussion	r/politics
Kobe Bryant	searches	subreddit	r/kobebryant
Tom Hanks	actors	subreddit	r/tomhanks
Joaquin Phoenix	actors	subreddit	r/joaquinphoenix
Amitabh Bachchan	actors	subreddit	r/india
Ryan Newman	athletes	subreddit	r/RyanNewman
Michael Jordan	athletes	subreddit	r/michaeljordan
Tyson Fury	athletes	subreddit	r/TysonFury
Parasite	movies	subreddit	r/parasite
1917	movies	subreddit	r/1917
Black Panther	movies	subreddit	r/blackpanther
Tiger King	tv shows	subreddit	r/TigerKing
Big Brother Brasil	tv shows	subreddit	r/BigBrotherBrasil
Money Heist	tv shows	subreddit	r/MoneyHeist
Joe Biden	people	subreddit	r/JoeBiden
Kim Jong Un	people	subreddit	r/kimjongun
Boris Johnson	people	subreddit	r/BorisJohnson
Coronavirus	news	subreddit	r/Coronavirus
Election results	news	discussion	r/politics
Iran	news	subreddit	r/iran
Among Us	games	subreddit	r/AmongUs
Fall Guys: Ultimate Knockout	games	subreddit	r/FallGuysGame
Valorant	games	subreddit	r/VALORANT
Dalgona coffee	recipes	discussion	r/cafe
Ekmek	recipes	discussion	r/Breadit
Sourdough bread	recipes	subreddit	r/SourdoughBread
Kobe Bryant	loss	subreddit	r/kobebryant
Naya Rivera	loss	subreddit	r/NayaRivera
Chadwick Boseman	loss	subreddit	r/ChadwickBoseman

Figures 5.2 and 5.1. The wordclouds are build from posts related to a subreddit dedicated (or closest) to the searched topic. *Reddit-TUDFE* allows to quickly check if, and how, a particular topic is covered. Note that similar examples can be derived for any other topic, while Reddit also shows potential in, for instance, building ontologies, or semantic graphs. However, this possibility is out of scope of this contribution.

In Figure 5.1:

- Left subfigure shows result for the phrase "music", a generic term, which is certainly discussed on Reddit. One may see particular genres: rock, pop, rap, relaxing, electronic, etc.
- Middle subfigure displays results for phrase "rock", a bit narrowed, but still vague music-related (sub)topic, which is also present in Reddit, including artists/bands like: Rolling Stones, AC/DC, Led Zeppeling, Queen, Pink etc.
- Right subfigure contains a strictly specific topic, i.e. the band "The Beatles", which is also widely covered on Reddit. Here one may see, among others, individual band members: John Lennon, Paul McCartney, Ringo Starr, and George Harrison.

Another example is summarized in Figure 5.2.

Fig. 5.1: Exemplary wordclouds of 200 posts (before 01-09-2021) concerning (top to bottom): “music”, “rock” and “The Beatles”.

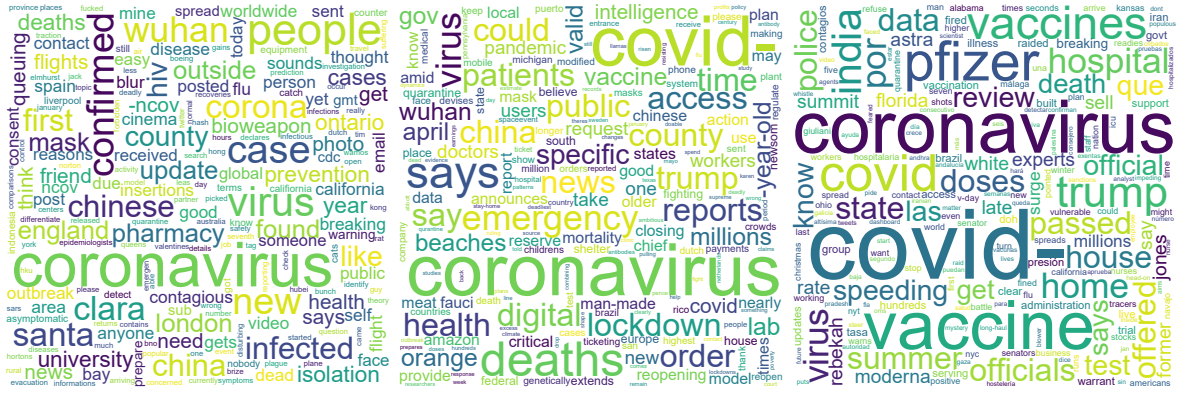

Fig. 5.2: Word clouds of 100 posts title from subreddit r/Coronavirus[39]at different times during COVID-19 pandemic (left to right: 01-02-2020, 01-05-2020 and 08-12-2020)

In this figure, one can notice a clear shift of focus on the subreddit r/Coronavirus at different times of COVID-19 pandemic [1] (phrase “coronavirus” is skipped). Figure 5.2 (left) (before 01-02-2020) shows that the main interest concerned the phrases a.o.: “confirmed” (the number of confirmed infections), “Wuhan” and “Chinese” (the geographical origins of the fist reported infections[25]). Figure 5.2 (middle) (before 01-05-2020) displays that the main phrases changed to: “deaths” (due to COVID-19 infection) and “lockdown” (the preventive measures against the spread of the virus). Figure 5.2 (right) (before 08-12-2020, i.e. near the first vaccine invention) shows the general interests in phrases like: “vaccine” and “pfizer” (the company to invent the vaccine [3]).

Note that analysing the evolution of thematic ecosystem is just one of possible applications of the *Reddit–TUDFE* tool. Most importantly, it quickly allows checking whether given topical domain contains live (evolving over time) information.

6. Concluding remarks. This work provides evidence that Reddit is a robust, but underutilized, resource for information retrieval and knowledge capture, in almost any field of interest. Based on performed exploratory analysis, the following answers to the research questions formulated at the beginning of this work can be stipulated:

[39]https://www.reddit.com/r/Coronavirus/

- **RQ1**: Reddit offers publicly available data, which can be easily retrieved with Pushshift API.
- **RQ2**: Most popular techniques for Reddit information processing are: text embeddings, neural networks, and graph networks.
- **RQ3**: Reddit is trending in scientific research as more and more articles using it are published every year.
- **RQ4**: Reddit covers the majority (79%) of topics that appear in Global Google Trends, sustaining the claim that Reddit is a robust source of knowledge about "everything trendy".
- **RQ5**: Reddit is most commonly used in tandem with Twitter.

These conclusions render Reddit a perfect candidate for future research – especially the presence of graph networks among common research methods and high coverage of popular trends. Finally, this analysis and the *Reddit–TUDFE* tool provide solid foundation for future research on Reddit and its potential in information retrieval.

Acknowledgement. This work has been supported in part by the joint research project "Novel methods for development of distributed systems" under the agreement on scientific cooperation between the Polish Academy of Sciences and Romanian Academy.

REFERENCES

[1] A. Abd-Alrazaq, J. Schneider, B. Mifsud, T. Alam, M. Househ, M. Hamdi, and Z. Shah, *A comprehensive overview of the covid-19 literature: Machine learning-based bibliometric analysis*, Journal of medical Internet research, 23 (2021), p. e23703.

[2] K. Aggarwal, P. Bamdev, D. Mahata, R. R. Shah, P. Kumaraguru, et al., *Trawling for trolling: A dataset*, arXiv preprint arXiv:2008.00525, (2020).

[3] A. Badiani, J. Patel, K. Ziolkowski, and F. Nielsen, *Pfizer: The miracle vaccine for covid-19?*, Public Health in Practice, 1 (2020), p. 100061.

[4] J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, and J. Blackburn, *The pushshift reddit dataset*, in Proceedings of the International AAAI Conference on Web and Social Media, vol. 14, 2020, pp. 830–839.

[5] C. Buntain, R. Bonneau, J. Nagler, and J. A. Tucker, *Youtube recommendations and effects on sharing across online social platforms*, Proceedings of the ACM on Human-Computer Interaction, 5 (2021), pp. 1–26.

[6] L. Cao, H. Zhang, and L. Feng, *Building and using personal knowledge graph to improve suicidal ideation detection on social media*, IEEE Transactions on Multimedia, (2020).

[7] M. Cinelli, G. D. F. Morales, A. Galeazzi, W. Quattrociocchi, and M. Starnini, *Echo chambers on social media: A comparative analysis*, arXiv preprint arXiv:2004.09603, (2020).

[8] V. Cox, *Exploratory data analysis*, in Translating Statistics to Make Decisions, Springer, 2017, pp. 47–74.

[9] A. Curtis, *The science of subjectivity*, Geology, 40 (2012), pp. 95–96.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805, (2018).

[11] A. Edelbo Lillie and E. Refsgaard Middelboe, *Danish stance classification and rumour resolution*, arXiv e-prints, (2019), pp. arXiv–1907.

[12] M. Fajcik, L. Burget, and P. Smrz, *But-fit at semeval-2019 task 7: Determining the rumour stance with pre-trained deep bidirectional transformers*, arXiv preprint arXiv:1902.10126, (2019).

[13] I. Garibay, T. A. Oghaz, N. Yousefi, E. C. Mutlu, M. Schiappa, S. Scheinert, G. C. Anagnostopoulos, C. Bouwens, S. M. Fiore, A. Mantzaris, et al., *Deep agent: Studying the dynamics of information spread and evolution in social networks*, arXiv preprint arXiv:2003.11611, (2020).

[14] J. E. Gray, M. C. Hamilton, A. Hauser, M. M. Janz, J. P. Peters, and F. Taggart, *Scholarish: Google scholar and its value to the sciences*, Issues in Science and Technology Librarianship, 70 (2012).

[15] M. Gusenbauer, *Google scholar to overshadow them all? comparing the sizes of 12 academic search engines and bibliographic databases*, Scientometrics, 118 (2019), pp. 177–214.

[16] H. Habib, M. B. Musa, F. Zaffar, and R. Nithyanand, *To act or react: Investigating proactive strategies for online community moderation*, arXiv preprint arXiv:1906.11932, (2019).

[17] G. Halevi, H. Moed, and J. Bar-Ilan, *Suitability of google scholar as a source of scientific information and as a source of data for scientific evaluation—review of the literature*, Journal of informetrics, 11 (2017), pp. 823–834.

[18] J. Hastings, O. Kutz, and T. Mossakowski, *How to model the shapes of molecules? combining topology and ontology using heterogeneous specifications*, in In Proc. of the Deep Knowledge Representation Challenge Workshop (DKR-11), K-CAP-11, Citeseer, 2011.

[19] D. M. E.-D. M. Hussein, *Analyzing scientific papers based on sentiment analysis*, Information System Department Faculty of Computers and Information Cairo University, Egypt, (2016).

[20] P. Jacsó, *Google scholar: the pros and the cons*, Online information review, (2005).

[21] P. Jacsó, *Google scholar revisited*, Online information review, (2008).

[22] J. Jasser, I. Garibay, S. Scheinert, and A. V. Mantzaris, *Controversial information spreads faster and further in reddit*, arXiv preprint arXiv:2006.13991, (2020).
[23] F. R. Jensenius, M. Htun, D. J. Samuels, D. A. Singer, A. Lawrence, and M. Chwe, *Benefits and pitfalls of google scholar*, PS: Political Science and Politics, (2018).
[24] V. Kharde, P. Sonawane, et al., *Sentiment analysis of twitter data: a survey of techniques*, arXiv preprint arXiv:1601.06971, (2016).
[25] K. Leung, J. T. Wu, D. Liu, and G. M. Leung, *First-wave covid-19 transmissibility and severity in china outside hubei after control measures, and second-wave scenario planning: a modelling impact assessment*, The Lancet, 395 (2020), pp. 1382–1393.
[26] E. Loper and S. Bird, *Nltk: The natural language toolkit*, arXiv preprint cs/0205028, (2002).
[27] E. D. López-Cózar, E. Orduña-Malea, and A. Martín-Martín, *Google scholar as a data source for research assessment*, in Springer handbook of science and technology indicators, Springer, 2019, pp. 95–127.
[28] E. D. López-Cózar, E. Orduña-Malea, A. Martín-Martín, and J. M. Ayllón, *Google scholar: the big data bibliographic tool*, Research analytics: boosting university productivity and competitiveness through scientometrics, (2017), p. 59.
[29] F. MacKellar, *Subjectivity in qualitative research*, EDUC 867 WEBSITE, (2012).
[30] M. A. Mannan, *Science and subjectivity: Understanding objectivity of scientific knowledge*, Philosophy and Progress, (2016), pp. 43–72.
[31] A. N. Medvedev, R. Lambiotte, and J.-C. Delvenne, *The anatomy of reddit: An overview of academic research*, in Dynamics on and of Complex Networks, Springer, 2017, pp. 183–204.
[32] I. Newton, *Philosophiae naturalis principia mathematica*, vol. 2, typis A. et JM Duncan, 1833.
[33] E. Orduña-Malea, J. M. Ayllón, A. Martín-Martín, and E. D. López-Cózar, *Methods for estimating the size of google scholar*, Scientometrics, 104 (2015), pp. 931–949.
[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research, 12 (2011), pp. 2825–2830.
[35] J. M. Perkel, *Why jupyter is data scientists' computational notebook of choice.*, Nature, 563 (2018), pp. 145–147.
[36] A. Rajaraman and J. D. Ullman, *Data Mining*, Cambridge University Press, 2011, p. 1–17, https://doi.org/10.1017/CBO9781139058452.002.
[37] B. M. Randles, I. V. Pasquetto, M. S. Golshan, and C. L. Borgman, *Using the jupyter notebook as a tool for open science: An empirical study*, in 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL), IEEE, 2017, pp. 1–2.
[38] Y. Shinyama, *Pdfminer: Python pdf parser and analyzer*, Retrieved on, 11 (2015).
[39] M. Shultz, *Comparing test searches in pubmed and google scholar*, Journal of the Medical Library Association: JMLA, 95 (2007), p. 442.
[40] G. Viglione, *How scientific conferences will survive the coronavirus shock.*, Nature, 582 (2020), pp. 166–168.
[41] S. Zannettou, *Towards understanding the information ecosystem through the lens of multiple web communities*, arXiv preprint arXiv:1911.10517, (2019).
[42] S. Zannettou, T. Caulfield, E. De Cristofaro, M. Sirivianos, G. Stringhini, and J. Blackburn, *Disinformation warfare: Understanding state-sponsored trolls on twitter and their influence on the web*, in Companion proceedings of the 2019 world wide web conference, 2019, pp. 218–226.
[43] H. Zhang, Z. Liu, C. Xiong, and Z. Liu, *Grounded conversation generation as guided traverses in commonsense knowledge graphs*, arXiv preprint arXiv:1911.02707, (2019).

Edited by: Dana Petcu
Received: Dec 22, 2021
Accepted: Apr 1, 2022


Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org

Volume 23, Issues 1, pp. 23–33, DOI 10.12694/scpe.v23i1.1969

INFORMATION RETRIEVAL AND DATA ANALYTICS IN INTERNET OF THINGS: CURRENT PERSPECTIVE, APPLICATIONS AND CHALLENGES

KRUTI LAVINGIA* AND RACHANA MEHTA†

Abstract. The Internet of Things has emerged as an evolving paradigm and has developed its presence in a variety of domains around us. The emergence of IoT has also emphasized the need to cater to challenges such as interoperability, smart IoT components adoption, authentication and authorization, networking, information retrieval, and several other issues. The ubiquitous nature and interconnection between various devices supported by machine learning, artificial intelligence, cloud computing, big data, and blockchain lead to a generation of large amounts of data. In order to find useful information from such data is a tedious task and involves high computation. The domain of Information Retrieval helps us to identify and manage environmental factors of data collected through sensors. The data gathered may be heterogeneous and from different sources. This demands the need for better retrieval efficiency, accuracy, and systematic models for gathering and managing the sensed data. Designing such a model with security and privacy is a major concern. The acquired knowledge from those models will be helpful for data analytics, performance-boosting, decision making, and managing the resources efficiently. A detailed study of the importance of Information Retrieval and Data Analytics in the Internet of Things is presented in this paper.

Key words: Internet of Things, Data analytics, Information retrieval, Big data, Data mining, Smart City, Machine Learning

AMS subject classifications. 68P20

1. Introduction.

The Internet of Things (IoT) has emerged exponentially in the past decade owing to the advances in wired and wireless devices in and around us [1]. Internet of Things involves the interaction and intercommunication among the connected heterogeneous devices [2]. The communication among those devices takes place using technologies such as Bluetooth, Global System for Mobile Communication (GSM), ZigBee, Radio Frequency Identification (RFID) sensors and actuators, WiFi, and several others. These technologies allow phones, laptops, gaming consoles, sensors to communicate with one another and exchange information [1, 2].

The impact of IoT technology on the Sustainable Development Goals (SDGs), devised by the United Nations (UN) has been a significant area of research. The current applications of IoT such as Smart Cities, Smart Learning, Smart Agriculture, Smart Healthcare, IoT based Intelligent Systems for Education, Water Management, Smart Grid, IoT in smart manufacturing, industrial automation are directly in line with the 2030 sustainability goals of the UN. A project initiative is taken by the World Economic Forum for the IoT for Sustainable Development, more than 640 projects are being analyzed for their applicability. The main focus is on developing scalable and reusable models which can be used by different stakeholders [7]. This signifies the need and importance of IoT-based solutions and their use.

IoT is prevalent and making its dominance in almost every field around us like home automation, logistics, manufacturing, healthcare, agriculture, smart grid, smart city, and much more [1, 3, 4, 5]. The wide adoption of IoT, the Internet, and the Industrial revolution like Web 4.0 has made a significant impact on the working culture and digitization. This has led to many companies going digital and leveraging the benefits of the online environment like working from anywhere, working at any time, accessing data from anywhere, increasing operational efficiency, reduction in infrastructural cost, and much more. Especially in the current scenario of COVID-19, this digitization has led the companies and industries to go and work smoothly. The boom in digitization and advancement of IoT has also floated up the challenges coming with it. Some of them are interoperability among the connected devices, adoption of the smartness among the IoT components, security

*CSE Department, Institute of Technology, Nirma University, India (kruti.lavingia@nirmauni.ac.in).
†CSE Department, Institute of Technology, Nirma University, India (rachana.mehta@nirmauni.ac.in).

and privacy concerns of authentication and authorization, issues pertaining to network and communication, retrieval of information, and analytics on data [1, 3, 6, 9].

The ubiquitous behavior and interconnection among various homogeneous and heterogeneous devices supported by new edge technologies like Artificial Intelligence, Machine Learning, Deep Learning, Blockchain, and Cloud Computing have led to a generation of large amounts of data. In order to fetch meaningful information from such a heap is a complex and time-crunching task, as well as involves a high degree of computation and resource utilization. This process of finding out useful information is known as Information Retrieval (IR). In the context of IoT, the IoT devices and sensors are the primary sources of data generation, this data may come in varied forms and changes according to the application at hand. Since the IoT is based on the intercommunication of networked devices, the IR task also gets affected by the network technology being used for a particular application. This brings in the new concern of efficient retrieval of data, devising the models that can gather sensor data, record data, and manage them using appropriate indexing mechanisms, addressing the security and ethical issues, and extracting information in real-time. On the other hand, the task of data analytics involves the work on data that is collected or retrieved. Data analytics perform various operations such as data cleaning, data preprocessing, data mining to extract the trends, patterns, hidden features, and other useful information present in the collected data. The analytics helps to know the working pattern for businesses, trends, and patterns which are, unless invisible, help to understand the dynamics of the working model and efficiently leverage them for futuristic work. Data Analytics in IoT follows different mechanisms and techniques for different applications. The task of data analytics is dependent on data used in the application, data generated in the application, and the approach used for analytics purposes. The current paper gives insights on the current work done in Information Retrieval and Data Analytics in IoT along with their issues and possible solutions.

The remaining part is divided as follows: Background provides the insights on IoT environment and what data analytics is, the next section gives a detailed overview of various use cases of IoT-based applications using data analytics, and the challenges pertaining to the data analytics are considered, followed by conclusions.

2. Background. Before we dwell on the concerns of Data Analytics and Information Retrieval using IoT, this section gives an understanding of the terminologies on hand.

2.1. IoT Environment. The Internet of Things involves the interconnection of various sensor-based devices and intercommunication among them with the help of varied communication technologies. The general IoT environment consists of mainly IoT devices, and networking technologies [8]. The IoT devices consist of sensors and actuators directly used or embedded within the system. Commonly used sensors include a temperature sensor, humidity sensor, pressure sensor, proximity sensor, infrared sensor, gas sensor, accelerometer, level sensor, gyroscope, motion sensor, chemical sensor, and much more. A single IoT device may consist of more than one sensor to measure multiple factors at a single point in time. The sensor and actuator-based devices are the primary sources of data collection. The data generated through sensors and actuators get stored locally either on the IoT device itself or on the external storage of the cloud, user system, or edge. The storage is purely dependent on the IoT environment. The data stored is further useful to the end-user. The same is transmitted to the end-user or shown with the help of the user interface. The IoT sensors are embedded, and along with it, the support for network communication technologies is done using the ports and antennas. The networking and communication technologies in the IoT environment involve support for the intercommunication among the IoT devices and from IoT device to end system. In most cases, the analysis of generated data is carried out to infer the notable trends, patterns, and features.

2.2. Data Analytics. The domain of data analytics involves the analysis of data to study the environment from where data is coming, the behaviour and working of the environment, to infer meaningful relationships, to make predictions, and to decide the future actions.The data generated from the sensor devices are taken as input for the data analysis and the outcomes of the analysis are patterns, hidden features, new information, and statistics of the system which helps in decision making. The volume of the data generated from IoT devices has increased manifolds owing to the increase in usage of smart devices, the Internet, advancements in communication networks, and IoT technologies in all domains. These involve using efficient data analysis tools which support heavy data, such as big data. The task of data analytics is not only limited to using the commer-

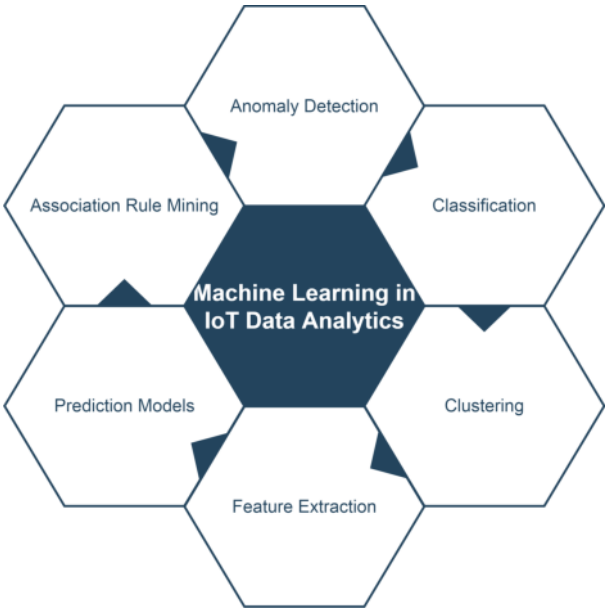

Fig. 2.1: Machine Learning Applications in IoT

cial data analytics software, but also includes using current edge technologies like Artificial Intelligence (AI). Machine Learning (ML), a sub field of AI and Deep Learning (DL), a subfield of ML are also prevelantly used for Data Analytics. These models use techniques like clustering, classification, prediction models, association rule mining, feature extraction, anomaly detection and many others for data analysis purpose. The current research focuses on the various ways data analytics can be carried out including, the analytics tools and various computing paradigms of AI, ML, and DL.

Fig 2.1 shows the various ML-based algorithms used for data analysis in IoT.

3. Use cases of IoT Data Analytics. There are a number of domain-specific applications in IoT that deal with huge amounts of data. For any such application, sensors are the key elements responsible for gathering information from the environment. The sensor-generated data is key to the data analysis in IoT applications. In this section of the paper, various IoT use cases for data analytics are discussed, provided that are helpful for the sustainability perspective.

3.1. Smart city. The smart city is one of the highly explored and leveraged IoT applications owing to its enormous benefits. The Smart city application covers all the IoT domains. It includes various IoT use cases like smart traffic management, smart waste management, smart governance, smart people, smart infrastructure, smart grid, smart people, smart healthcare, smart sanitation, smart monitoring, smart environment, smart economy, and much more.

Rathore et al. [10] have proposed a prototype of Smart Digital City along with the big data processing on a real-time basis. Apache Spark over Hadoop has been taken into consideration for data analysis on smart city applications like home, parking, climate, pollution monitoring, and vehicle network. The analyzed data is used by the government and municipalities for making decisions such as traffic analysis, diverting the traffic when there is congestion, showing the empty parking spots in a city, city planning based on the pollution status in a given area, and managing the water resource using the smart home data [10]. Jinping Chang et al [11] propose an adaptive heuristic mathematical model for traffic congestion. The model work includes monitoring the city traffic through sensors in all day and all-out emissions. The collected data is further sent to the central place which organizes the traffic police efficiently and sustainably. As well the traffic can be diverted to other paths, leading to reduction of carbon emission at specific places. The proposed model includes video monitoring,

surveillance, GPS location tracking system, and GIS to generate the data. The data is further analyzed using the traffic status, alerts, and graph to divert the traffic and manage it appropriately. M. Ashwin et al [12] have devised an automated intelligent smart bin to resolve the waste management issue. Ultrasonic sensor and servo motor are used to detect the human presence and overall work of the bin respectively. Furthermore, the capacitive sensor is used to segregate the dry and wet waste. As well it works on the solar panel to run the bin efficiently without any external energy source. An optimal route selection algorithm is used to dispose of the bin and install the empty one.

3.2. Smart Building. The smart building stands at the core of the infrastructural aspect of a smart city. Smart buildings are useful for the effective management of a building, its construction, monitoring the progress of residents and their concerns, home automation, real-time alert management, intrusion detection in surveillance of the building, efficient utilization of energy, waste management, individual authorization, etc. Smart buildings are concerned with using the IoT at every level from architectural planning, construction to implementation and final results.

M.Dey et al [13] have proposed a novel feature extraction technique for extracting the power and temperature information from the high dimensional terminal unit data of Heating Ventilation and Air-Conditioning (HVAC). An unsupervised X-means clustering method has been applied to find the faulty unit and in the second stage, Multi-Class Support Vector Machine (MC-SVM) algorithm has been applied to find the pattern of fault detection and provide an appropriate diagnosis. This helps in reducing the amount of energy wastage. Also, an automated alert is employed to inform the maintenance team. Isaac et. al. [14] have proposed the HEMS-IoT, a framework for energy management in smart homes, comfort, and safety. The collected sensor data is analyzed using the J48 classification algorithm and WEKA tool. The analysis provides insights on the user behavior and the energy consumption by the home devices. Further RuleML and Apache Mahout have been used to provide energy saving and safety recommendations to the user. Wei Zhang et al [15] have devised the thermal comfort model for smart buildings. Machine Learning-based approach is used to design the model, it helps in bridging the gap between the controllable parameters of building and thermal comfort. Neural Network-based models have been found efficient for the data analysis and control parameter tuning using the real-time data for time, date, and weather. M.R. Bashir [16] has proposed an integrated IoT big data analytics framework for the real-time analysis of oxygen levels, luminosity, and hazardous gas in different parts of the building. The IoT sensors incorporated include the oxygen sensor, gas detector, and luminosity sensor. The real-time data generated is fed to the PySpark for the analysis purpose, and Cloudera for visualization purposes, which alerts for turning on the oxygen pump, when the oxygen level drops, and similarly for turning off the pump when it is within a threshold value. The data generated from the gas sensor is used for alerting and tuning the fire alarm. A luminosity level sensor is used to turn on and turn off the lights. This helps in the efficient management of energy and smart control of buildings.

3.3. Smart Agriculture. The Internet of Things helps in dealing with a number of challenges related to practical farming. The modernization in the field of Agriculture is taking place due to the advancements in the IoT systems. A variety of agriculture-related issues such as identification of temperature and climate beneficial for the crop, temperature, and productivity of the soil, water level required, horticulture, usage of pesticides, manure requirement, etc can be identified using different sensors and these tasks can be automated using IoT sensors and applications. Lavanya G. et al.[17] have designed a novel sensor called the NPK Sensor (Nitrogen, Potassium, Phosphorus Sensor) which aids in monitoring and identifying the deficiency of nutrients present in the soil. This sensor uses the colorimetric principle to carry out analysis on the nutrients that are present in the soil. It has light-emitting diodes and Light-dependent resistors in it. The data that is sensed by these NPK sensors are then forwarded to the cloud and fuzzification logic is applied to it as the data retrieved is in a very vague format. From the analysis carried out, an alert message is sent to the farmers to inform them the quantity of nutrients in the form of N, P, and K required by the soil at regular intervals. This helps in yielding good quality and quantity of crops. Kang, Ju-Hee et al. [18] have designed and implemented an information retrieval system that aids in retrieving information related to insect pests and diseases for the cultivation of crops and helps the users in checking real-time information with the help of their phones using Lucene which is a library that specifically works on image analysis.

3.4. Smart Transportation. With the advancement of sensors, communication technologies, automated and high-speed networking technologies, transportation and traffic management in cities have become quite modernized and smarter. Smart transportation helps in enhancing the efficiency of an individual in moving around from one place to another along with ensuring safety. The technologies facilitating this smart transportation include IoT devices and the 5G networking technology. Yongming Feng in [19] proposes a solution faced by vehicular networks in real-time information retrieval of data related to vehicular navigation and its positioning. The traditional method of information retrieval is not that efficient in fetching real-time data. The proposed solution on the analysis of the Frame difference method, the optical flow method, and the Background difference method plays a major role in the detection of moving vehicles or targets. The pros and cons of all these three methods are discussed. The author mainly focuses on the study of image features-based vehicle retrieval algorithms for proposing a solution for information retrieval of the navigation of vehicles and their positioning. Camilo Castellanos et al. [20] using different case studies represents a RA for addressing the challenges of big data analysis in STS that includes architectural patterns and different tactics.

3.5. Smart Industry. Jing Wang et al in [21] have proposed a pre-warning system for food safety that adopts association rule mining and IoT technology, for monitoring at regular intervals of time, all the detected data of the entire supply chain and automatically pre-warning the system if any abnormality is detected. K. Moorthi et al. [22] have carried out data analytics for various e-commerce companies using historic and static data. The analysis proves that data grows and changes now and then and it is a requirement of new models as well as algorithms for collecting, storing, processing, analyzing, and evaluating the data in any e-commerce related applications.

3.6. Other Case Studies. Irfan Mehmood et al. in [23] propose a solution to deal with the computational quality and storage-related challenges faced during image retrieval through smartphones in an IoT environment. The authors have proposed a deep learning-based lightweight system for energy-constrained devices. The steps mentioned in the proposed system are first detection and cropping of face regions using classifiers. Then using convolutional layers to represent faces, then indexing the big data repository, for faster comparison for real-time retrieval, and finally using Euclidean distance for finding a resemblance between the images in repositories and the concerned queries. Navjyot Kaur Walia et al. in [24] have mentioned the use of information retrieval in designing an IoT-specific application for controlling Smart Lights using Wifi. Mingliu Liu et al. in [25], after investigation of a number of IoT searching scenarios, have proposed a common model for representing recordings of sensor information. The authors propose an Indexing mechanism and a tree indexing structure for improving the efficiency and accuracy of the retrieval process and ensuring the scalability for any large-scaled data simultaneously. A large number of simulations were also carried out for demonstrating the effectiveness of their proposed solution. Ananda Ghosh et al. in [26] propose an approach based on deep learning for the reduction of data on the edge with machine learning on the cloud by investigating the merging of edge and cloud computing for IoT data analytics. To reduce the dimensions of data, the auto-encoder's encoder part is placed on the edge which helps in reducing the dimensions of data. The reduced amount of data is then transferred to the cloud from where it can be directly used for machine learning. The data can be also converted back to its original features by the auto-encoder's decoder part. The approach proposed was evaluated and the results show that the reduction of data did not have a major impact on the classification accuracy. Only a minor effect was seen on the classification accuracy even by a 77% reduction in data.

Table 3.1 summarizes the various IoT applications, data used in that application, IoT device or sensors and data analytics tool being used.

Table 3.1: IoT Data Analytics Applications

Source	**IoT Application**	**Type of Data**	**Summary**	**IoT Device**	**Year**	**Data Analytics Tool**	**Pros**	**Cons**	**Future work**	**Methodology used**
[10]	Smart City	Real Time Text	Real Time analysis of urban city data	ZigBee based Vehicular Network	2018	Apache Spark, Hadoop, Giraph	Efficient in terms of Scalability and Big Data Processing	Time complexity grows with amount of data	Integration of graph network and dynamicity	Graph Algorithms using Mapreduce
[11]	Smart City	Real Time Data	Adaptive heuristic model for traffic congestion and carbon emission	Signal processing and location based sensor	2020	Data Graph	Comprehensive Design for Smart Traffic and Congestion forecasting, Efficient Accuracy and less error rate	Societal and demographic activity factors not included like major events	Intelligent Transport Services, Driverless Technology can be mapped with IoT Devices	Probability Analysis Model
[12]	Smart City	Real Time Data	Solar power based smart bin to segregate wet or dry waste and overflow alert system	Ultrasonic Sensor, Level Sensor, weight sensor	2020	Data Graph	Dry and wet waste segregator, odour control, human detection facility, route mechanism	Solar panel battery and regular maintenance	Autonomous movable smart bin	Optimal route selection algorithm is used for route selection for bin disposal
[13]	Smart Building HVAC Fault Detection and Diagnosis	Real Time Data	Feature Extraction for HVAC fault finding and alert	Terminal Unit data of heating, cooling, deadband	2018	Apache Spark, Cassandra and MC-SVM	Effective fault diagnosis for all faulty and non faulty terminal units of HVAC	Model is application for fan coil terminal unit of single building, Non inclusion of internal and external faults	Fully remote fault diagnosis, Generalized terminal unit model can be developed	X-Means Clustering, Multi Class Support Vector Machine

Source	IoT Application	Type of Data	Summary	IoT Device	Year	Data Analytics Tool	Pros	Cons	Future work	Methodology used
[14]	Smart Building Smart Home Energy Management	Real Time Data	Energy Management for a Smart home	Home Automation Sensor	2020	J48, WEKA	Recommendation for smart home management, Demographic specific energy saving results	Application is specific to android domain only, customized energy saving options can not be added, Limited devices were considered	Model validation on many devices, Incorporation of location based devices, Incorporation of blockchain and cyber security	J48 Machine Learning algorithm for energy consumption pattern and behaviour, RuleML for energy efficient recommendation
[15]	Smart Building Thermal Comfort	Real Time Data	Optimal parameter setting for thermal comfort	Humidity and Temperature Sensor	2018	Nonlinear ML models (NN)	Better performance of non linear models compared to linear ones with minimal training time	In addition to HVAC settings, other factors impacting building energy usage should be investigated.	Other data aspects, such as occupancy and building location can be incorporated	Machine learning techniques for comfort level modelling
[16]	Smart Building Smart Control	Real Time Data	Smart control of oxygen level, smoke detection and luminosity level in building	Oxygen Sensor, Gas Detector and Luminosity sensor	2016	PySpark and Cloudera	Integration of BDA and IoT for handling the enormous volume and velocity challenge of real-time data in the smart building area	Applicable only on the Smart building domain	Applicability to other domains	Python and the Big Data Cloudera platform

Source	IoT Application	Type of Data	Summary	IoT Device	Year	Data Analytics Tool	Pros	Cons	Future work	Methodology used
[17]	Smart Agriculture Fertilizer Intimation System	Voltage based on the chemicals present in the soil	Design of a Nitrogen Phosphorus Potassium sensor.	Novel NPK Sensor	2020	Google cloud with fuzzy logic	IoT solution that is low-cost, accurate, and intelligent	The sms generated do not mention the amount or quantity of fertilizer to be added	A separate module can be added to the system mentioning the required amount of fertilizer	Fuzzy Rule based system
[18]	Retrieving information related to insect pests and diseases for cultivation of crops for u-Farm	Images of the farm	Diseases and Pest analysis is carried out and users can monitor the analysis at real time that helps in better yielding of crops	High resolution cameras	2015	Lucene library for image	Realtime information retrieval on smartphones	Could include more functionalities such as temperature humidity factor monitoring	More functionalities such as modules on irrigation can be integrated on the uFarm app	Object oriented modeling
[19]	Vehicular Navigation and Positioning	Image features based vehicle retrieval	Focus is on image features based vehicle retrieval algorithms for proposing a solution for information retrieval needed for Vehicular Navigation and Positioning	Transient phone signals, GPS Trajectory	2020	Frame Difference Method, Optical Flow method and Background difference method	Efficient real-time performance with better detection capabilities.	Absolute precision in vehicle navigation is difficult and retrieval of positional data.	The method's data bytes are very close to the actual bytes and as time passes they will essentially coincide with the actual bytes	Image Matching

Source	IoT Application	Type of Data	Summary	IoT Device	Year	Data Analytics Tool	Pros	Cons	Future work	Methodology used
[20]	Addressing Big Data Analysis in STS	Architectural Patterns and different Tactics	Represents a RA that uses different case studies for addressing BDA	RA	2021	RA for addressing the challenges of big data analysis in STS	Proposes architectures for both Big Data Analytics and Smart Transportation System	Limited capabilities over large size of data	Focus on analysis to be carried on voluminous data	Architectural patterns and tactics
[21]	Sustainable Food Supply Chain	Data for entire supply chain	Data for the supply chain is collected and pre-warning is generated if any abnormality is detected	No Specific Tool	2017	Association Rule Mining	Effective identification of safety related issues	Only limited case studies identified and worked upon	More case studies related to the supply chain can be considered	Association rule mining
[23]	Quality and storage related challenges in Image retrieval	Images from Smartphones	Solution that uses Deep learning based light weight system to solve the quality and storage related challenges during Image retrieval from smartphones	Image extraction from Smart Phones	2019	Deep Learning based light-weight system	Approach is extremely efficient both in terms of complexity and accuracy	Features still cannot be stored on small capacity devices and performance of retrieval should be robust	Analyzing image representation techniques that are based on hashing	Binary Classifier and CNN Feature extraction

4. Challenges in IoT Data Analytics. In this section we present the open issue and challenges faced in IoT based Data Analytics.

4.1. Data Acquisition and Transmission. The data generated by the IoT sensors are of different types, having different structures and different dimensions. IoT Sensor data can be structured, semi-structured, or unstructured. It can be homogeneous or heterogeneous in nature. The generated data further needs to be transmitted to the database or cloud space. To transfer such large, complex, and dynamic data requires the appropriate transmission protocol, relevant to the application on hand. For some applications, the amount of data generated varies from time to time, in the case of smart traffic, the data is generated at every fraction of a second, whereas in smart agriculture the data is generated on a daily or weekly basis. The data storage needs to be appropriately chosen which can handle the data according to application, supports scalability and heterogeneity of data. In the scenario of data analytics, data access time is also important, data storage should be able to provide fast and efficient access to the stored information.

4.2. Data Processing. Prior to performing the data analytics, the data needs to be cleaned to remove the noisy and erroneous data, remove redundant data, perform the integration of the data generated from different sources, data conversion to the form as needed for data mining, removing the data abnormalities and analysis which includes techniques like normalization, scaling. If the data is voluminous, the analysis will be delayed and might not be useful for the dynamic application. The data generated by the IoT sensors may be mapped with one another in the temporal or spatial domain, it is important to learn and know that prior to analysis. The data analysis and information retrieval can be sequential or parallel in nature, considering the amount of generated data, if it is voluminous and independent, parallel data analysis is useful. While if the data needs the time ordinance, sequential processing is useful. So it is necessary to consider the data characteristics and apply the data pre-processing techniques before performing the data analysis.

4.3. Security and Privacy. The IoT Data Analytics involves data to be transferred to either the third-party data storage or the analytics platform. The data in our IoT system can span over various categories like belonging to demographics of a person, sensitive information of a government, medical information of a patient, location of a transport vehicle, and much more. As the data moves out of the IoT ecosystem, it is a prime concern for data to maintain the integrity and be secure from unauthorized access and malicious use. This requires taking the appropriate handling mechanism like service level agreement and protocol, before sending the data. Before sending data over cloud or third party, it is necessary to check these things.

4.4. Data Analytics . The data once reached the cloud or third-party software or in the internal system, requires the correct analysis to be performed. The data analytics tool or method should be selected such that it can handle the capacity, variety, and dynamic nature of data or information. The result generated from the analysis should be properly represented and visualized for better understanding. The tool or method used for analysis must be time efficient and provide an accurate result. The tool should be robust and work in all scenarios. In the current scenario, machine learning and deep learning techniques have proliferated in the domain of data analytics. Selecting the ML or DL techniques requires an understanding of the domain. They come with their challenges of time versus accuracy tradeoff, parameter tuning, efficiency, and much more. All these things should be considered, before selecting the data analytics tool or techniques.

5. Conclusion. Data analytics and information retrieval are important for any application to infer the working of the system, to make a future decision or predict the working of the model, or to infer trends, patterns, and many more hidden features. Static data once achieved from the IoT sensor if not used for analysis purposes, remains meaningless. In this paper, the overview of how data analysis is used in various IoT applications is discussed, along with potential tools and techniques used for analysis. Further on, the issues and challenges are also discussed for future exploration.

REFERENCES

[1] Atzori, L, Antonio I, and Giacomo M., *The internet of things: A survey.*, Computer networks 54, no. 15 (2010): 2787-2805.
[2] Yaqoob, I, Ejaz A, Ibrahim A T H, Abdelmuttlib I A A, Abdullah G, M I, and Mohsen G.,*Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges.*, IEEE wireless communications 24, no. 3 (2017): 10-16.

[3] Bandyopadhyay, D, and Jaydip S,*Internet of things: Applications and challenges in technology and standardization.*, Wireless personal communications 58, no. 1 (2011): 49-69.
[4] Ou, Q, Yan Z, Xiangzhen L, Yiying Z, and Lingkang Z. ,*Application of internet of things in smart grid power transmission.*, In 2012 third FTRA international conference on mobile, ubiquitous, and intelligent computing, pp. 96-100. IEEE, 2012.
[5] Khanpara P, Lavingia K. ,*Energy conservation in multimedia big data computing and the Internet of Things—A challenge In Multimedia Big Data Computing for IoT Applications*, 2020 (pp. 37-57) Springer, Singapore.
[6] Mendez, D M., Ioannis P, and Baijian Y. ,*Internet of things: Survey on security and privacy.*, arXiv preprint arXiv:1707.01879 (2017).
[7] IoT for Sustainable Development Project, *Widgets.weforum.org*, 2021. [Online]. Available: https://widgets.weforum.org/iot4d/index.html. [Accessed: 20- Oct- 2021].
[8] Elijah, O, Tharek A R, Igbafe O, Chee Y L, and MHD N H.,*An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges.*, IEEE Internet of Things Journal 5, no. 5 (2018): 3758-3773.
[9] Chormunge S, Mehta R. ,*Comparison Analysis of Extracting Frequent Itemsets Algorithms Using MapReduce*,In Intelligent Data Communication Technologies and Internet of Things 2021 (pp. 199-210). Springer, Singapore
[10] Rathore, M. M, Anand P, Won-Hwa H, HyunCheol S, Imtiaz A, and Sharjil S. ,*Exploiting IoT and big data analytics: Defining smart digital city using real-time urban data.*, Sustainable cities and society 40 (2018): 600-610.
[11] Chang, J, Seifedine N K, and Sujatha K. ,*Review and synthesis of Big Data analytics and computing for smart sustainable cities.* , IET Intelligent Transport Systems 14, no. 11 (2020): 1363-1370.
[12] Ashwin, M., Abdulrahman S A, and Azath M. ,*IoT based intelligent route selection of wastage segregation for smart cities using solar energy.*, Sustainable Energy Technologies and Assessments 46 (2021): 101281.
[13] Dey, M, Soumya P R, and Sandra D. ,*Smart building creation in large scale HVAC environments through automated fault detection and diagnosis.*, Future Generation Computer Systems 108 (2020): 950-966.
[14] Machorro-Cano, I, Giner A-H, Mario A Paredes-V, Lisbeth R-M, José L S-C, and José O O-A. ,*HEMS-IoT: A big data and machine learning-based smart home system for energy saving.*, Energies 13, no. 5 (2020): 1097.
[15] Zhang, W, Fang L, and Rui F. ,*Improved thermal comfort modeling for smart buildings: A data analytics study*, International Journal of Electrical Power and Energy Systems 103 (2018) : 634-643
[16] Bashir, M R, and Asif Q G. ,*Towards an IoT big data analytics framework: smart buildings systems.*, In 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 1325-1332. IEEE, 2016.
[17] Lavanya, G., Chellasamy R, and Pugalendhi G. ,*An automated low cost IoT based Fertilizer Intimation System for smart agriculture.*, Sustainable Computing: Informatics and Systems 28 (2020): 100300.
[18] Kang, J-H, Se-Hoon J, Sun-Sik N, Won-Ho S, and Chun-Bo S. ,*Design and implementation of produce farming field-oriented smart pest information retrieval system based on mobile for u-Farm.*, The Journal of the Korea institute of electronic communication sciences 10, no. 10 (2015): 1145-1156.
[19] Feng, Y. ,*Real Time Retrieval Technology of Vehicle Navigation and Location Information under Internet of Things Environment.*, In 2020 International Conference on Advance in Ambient Computing and Intelligence (ICAACI), pp. 185-188. IEEE, 2020.
[20] Castellanos, C, Boris P, and Dario C. ,*Smart Transportation: A Reference Architecture for Big Data Analytics.*, In Smart Cities: A Data Analytics Perspective, pp. 161-179. Springer, Cham, 2021.
[21] Wang, J, and Huili Y. ,*Food safety pre-warning system based on data mining for a sustainable food supply chain.*, Food Control 73 (2017): 223-229
[22] Moorthi, K., Gaurav Dhiman, P. Arulprakash, C. Suresh, and K. Srihari. ,*A survey on impact of data analytics techniques in E-commerce.*, Materials Today: Proceedings (2021).
[23] Mehmood, I, Amin U, Khan M, Der-Jiunn D, Weizhi M, Fadi A-T, Muhammad S, and Victor H C. de Albuquerque. ,*Efficient image recognition and retrieval on IoT-assisted energy-constrained platforms from big data repositories.*, IEEE Internet of Things Journal 6, no. 6 (2019): 9246-9255.
[24] Walia, N K, Parul K, and Deepti M. ,*An IoT by information retrieval approach: Smart lights controlled using WiFi.*, In 2016 6th International Conference-Cloud System and Big Data Engineering (Confluence), pp. 708-712. IEEE, 2016.
[25] Liu, M, Deshi L, Qimei C, Jixuan Z, Kaitao M, and Song Z. ,*Sensor information retrieval from Internet of Things: Representation and indexing.*, IEEE Access 6 (2018): 36509-36521
[26] Ghosh, Ananda M., and Katarina G. ,*Deep learning: Edge-cloud data analytics for IoT.*, In 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), pp. 1-7. IEEE, 2019.

Edited by: Katarzyna Wasielewska
Received: Jan 19, 2022
Accepted: Apr 6, 2022


Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2022 SCPE. Volume 23, Issues 1, pp. 35–50, DOI 10.12694/scpe.v23i1.1971

A COMPREHENSIVE SURVEY ON ENERGY CONSUMPTION ANALYSIS FOR NOSQL

MONIKA SHAH*, AMIT KOTHARI†, AND SAMIR PATEL‡

Abstract. During the last few years, we are witnessing increasing development in the Internet of Things (IoT) and big data. To address increasing workload complexity with better performance and to handle scalability issues of such applications, non-relational (NoSQL) has started taking the place of relational databases. With increasing load, it is challenging to maintain NoSQL's performance, scalability, and availability without expanding the capacity of hosts and power budget of computing resources [57]. Future scaling of data center capabilities depends on the improvement of server power efficiency [22, 33]. Considering the rise of energy costs and environmental sustainability, we can not ignore this high energy consumption caused by NoSQL. Despite the increasing popularity and share of NoSQL in the software market, little is still known about its energy footprint. To the best of our knowledge, there are no comprehensive studies that analyze the energy consumption by various modules of NoSQL. This article, therefore, conducts a comprehensive survey on the energy consumption analysis of NoSQL. There are limited proposals to reduce the energy consumption of NoSQL. This paper also provides a brief description of these little efforts on reducing the energy consumption of NoSQL. Based on the review, this paper discusses the research scope and opportunities for researchers to improve the energy conservation of NoSQL systems.

Key words: NoSQL, Energy Consumption, Efficiency Analysis, Energy Conservation, Power Management, Proportionality, Trade-off Analysis, power distribution

AMS subject classifications. 68M20, 97P30

Acronyms.

ACID	Atomicity, Consistency, Isolation, and Durability
BASE	Basically available, soft-state, and Eventually consistent
CPU	Central Processing Unit
DRAM	Dynamic Random Access Memory
DVFS	Dynamic Voltage and Frequency Scaling
EC	Energy Consumption
EE	Energy Efficiency
IoT	Internet of Things
LCS	Leveled Compaction Strategy
NoSQL	Not only SQL (Non-relational Database)
RAPL	Running Average Power Limit
RDBMS	Relational database management system
SLO	Service Level Objective
STCS	Size Tiered Compaction Strategy
TPC-H	Transaction Processing performance Council
WEC	Waiting Energy Consumption
YCSB	Yahoo! Cloud Serving Benchmark

1. Introduction.

It is not easy to imagine human life without the internet, computers, and mobile applications in this modern era. Online services like e-commerce, online banking, and social networking have become part of our daily routine. Easy access and reducing the cost of internet access have attracted developers to

*Computer Science and Engineering Department, Nirma University, India (monika.shah@nirmauni.ac.in).
†Gujarat Technological University, India (amitdkothari@gmail.com)
‡Computer Science and Engineering Department, Pandit Deendayal Petroleum University, India (samir.patel@sot.pdpu.ac.in)

expand these services using cloud computing and IoT. With the expansion of such applications wondering across the world today, the complexity and dimensions of data transmitted are growing exponentially with each passing year. For 30 years, database software has advanced to deal with these challenges. NoSQL database covers the shortage of traditional databases [35] while widely used as Big Data storage recently [39]. A wide class of NoSQL databases is available to meet different applications' requirements. Researchers of NoSQL databases are still struggling to optimize their performance with the increased scalability and complexity of data. Generally, the NoSQL database executes over a distributed cluster system to support horizontal scalability, where the NoSQL database schedules jobs scheduled to different nodes of a given cluster. The designing perspectives of NoSQL databases and relational databases are different. Relational databases are popular for transactional applications, where updates and Delete are the most frequent operations. At the same time, NoSQL databases' main perspective is handling massive data records and availability. In NoSQL databases, Create and Read operations are most popular, and operations like Update and Delete are replaced through ″timestamp″or ″data version″.

A study presented in [2] shows past and projected world energy consumption, which highlights a continuous rise in energy consumption. Database, analytics, and IoT will be the fastest-growing applications [1]. Statistics given in [22] tells 71% of data centers are occupied for big data processing. Data centers are known to be energy-hungry infrastructure running internet-based services [12, 64]. The work depicted in the study [26] has also warned about the need to reduce energy consumption at data centers. As a result, there are continuous efforts instituted on hardware-level and operating system-level energy management of data centers [12, 64]. Simple models that work well for hardware may not work well for software [12]. Due to a lack of application information (like resource consumption, data access pattern, etc. [62], the operating system also becomes inadequate to provide a pro-active energy-aware solution. Database workload is different than other workload [44]. In addition, the database is such an exceptional application, which can expose alternate execution plans in advance. It is also reported that power consumption by back-end database services is higher than front-end web services [49]. Power consumption in the database system was evidenced well at all stages of development [3]. Therefore, modern databases are posing high requirements on energy efficiency in addition to other metrics [36]. Many researchers have applied energy-aware processing of relational database systems [29, 19, 51, 23, 32, 31]. Energy consumption of NoSQL database is the average electricity consumed by computing nodes of NoSQL clusters for executing some tasks. Despite the increasing share of NoSQL in the database world, a lack of energy consumption model has been observed. With the growing load on the data center, it is becoming difficult to maintain the NoSQL application's performance without both increasing the processing capacity of hosts and increasing the power budget of computing resources [57]. Increasing power consumption by NoSQLs and the trend of power economic development shows research direction to analyze and reduce power consumption by NoSQL.Energy consumption an

The major contribution of this comprehensive survey can be summarized as follow:

- Power distribution among various components of NoSQL servers.
- Classifying ECA of NoSQL databases to its modules like query processing, query optimization, data modeling, and configuration of cache structure, cloud patterns, consistency levels, and latency.
- Study of energy proportionality to understand energy efficiency scope and trade-off requirement.
- Summary of proposed energy conservation techniques.
- Analyze research scope to optimize the energy consumption of NoSQL systems.

The rest of the paper is organized as follows. Section 2 shows power distribution among various components like CPU, and memory. It also provides a summary of power monitoring tools used to analyze the energy consumption of NoSQL databases. Section 3 is the heart of the article, which provides a comprehensive survey on energy consumption analysis of various NoSQL functional modules. Section 4 discusses the proportionality study of energy with workload, performance, and latency. It helps to identify trade-off requirements between energy conservation and other metrics. Minimal efforts are found to reduce the NoSQL system's energy consumption. Section 5 describes these efforts. Based on the review, gaps, challenges, and some directions for research in energy conservation of NoSQL are discussed in section 6, and conclusion in Sect. 7. Fig. 1.1 demonstrates an overview of Energy Consumption Analysis of NoSQL.

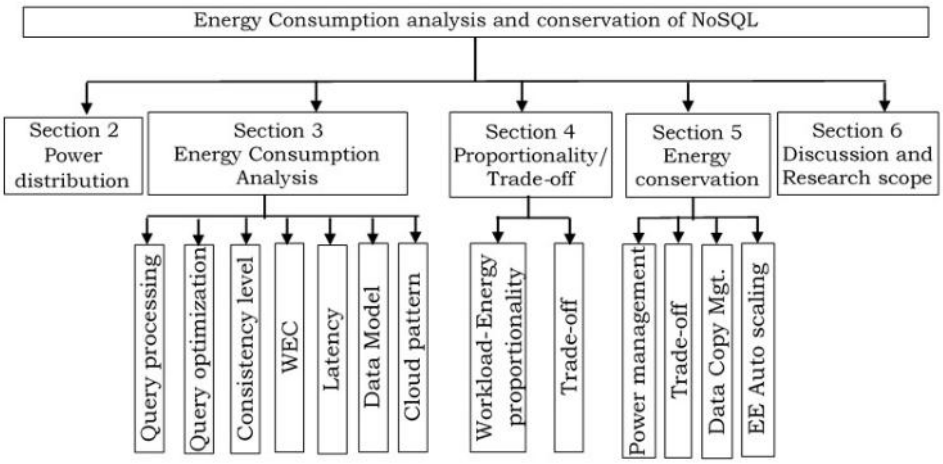

Fig. 1.1: An overview of Energy Consumption Analysis of NoSQL

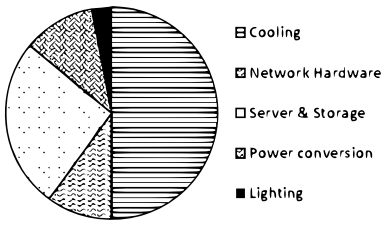

Fig. 2.1: Data Centre Power distribution

2. Power Distribution. Reliable measurement for each component of the system is an essential step toward sustainable energy consumption [11]. It helps to identify where power goes.It helps identify where power goes, and it can be helpful to researchers to identify components with unnecessary power supply and excessive power consumption. Therefore, this section provides a comprehensive survey on power distribution. Energy consumption can be presented as the product of power and execution time.

Fig. 2.1 shows energy consumption by different components of a data center, where the cooling system consumes more energy [12]. In a typical data center server, storage and network devices consume around 40%, 37%, and 23% of the total IT power, respectively [39]. 1 Watt of IT power saving can reduce 2.5 Watt in total power [39]. Data provided by Intel labs in [40], and the energy consumption survey of the data center presented in [12] reveal that a significant fraction of power consumed by a server is accounted for by the CPU, followed by the memory. NoSQL databases primarily use cluster setup. The component-level power distribution of the Cassandra cluster for the read-only and update-only workload is presented in [55, 54], which demonstrates two critical observations: i) In the Idle state, the highest power consumption is of components other than CPU and Memory. ii) The processor package adds the highest power consumption in the Active state. Static and dynamic power consumption analysis of in-memory databases presented in [28] also shows that dynamic power consumption of CPU and memory increases from 18% to 82%, and concludes that processor and memory consume significant energy during execution. Therefore, most NoSQL energy consumption analysis focuses on power consumed by CPU and Memory. Table 2.1 summarizes the components for which energy consumption is measured by literature work related to the energy consumption analysis of NoSQL.

Energy measurement can be done in three ways:i) Hadware-based, ii)Software-based, and iii) Hybrid. Hardware-based energy measurement has disadvantage of expensive and complex setup [11]. Reducing the

Table 2.1: Power consumption monitoring for components

Reference	CPU/ Processing Unit	DRAM /Memory	Entire System
[6]	✓	✓	
[14]	✓	✓	
[18]			✓
[34]			✓
[36]	✓	✓	
[37]	✓	✓	
[52]	✓	✓	
[53]	✓	✓	
[54]	✓		
[55]	✓		
[57]			✓

Table 2.2: Power monitoring tools

Reference	PowerAPI	jRAPL	Intel's RAPL	Power-meter	Other
[6]	✓				
[14]		✓			
[17]					API
[18]					Emeter software and EVM430-F6736 hardware
[34]				✓	
[36]					Log_power_to_file API
[37]					Log_power_to_file API
[52]		✓			
[53]	✓				
[54]			✓		
[55]			✓		
[57]				✓	

energy consumption of the NoSQL database system is a challenging task. The first challenge is to analyze energy consumption by different functional modules of NoSQL systems. Identifying suitable tools to measure the energy consumption of the NoSQL cluster is another challenge. Table 2.2 presents different approaches adopted by researchers to measure the energy consumption of NoSQLs.

Intel's Running Average Power Limit (RAPL) is a powerful tool that uses a software power model to estimate the energy consumption with the help of hardware performance counters and I/O models. It works for Intel processor architectures of Skylake, Haswell, Sandy Bridge, and Ivy Bridge. It provides accurate energy reading for CPU and RAM [25]. jRAPL is an API in java to monitor the energy consumption using RAPL. PowerAPI uses RAPL counters to provide power consumption information of each socket of the monitored machine [13]. In our literature survey, most articles use RAPL directly or through PowerAPI to measure the energy consumption of CPU and RAM components. To measure the energy consumed by the entire system, different meters like power-meter and Emeter are used in some research. Mahajan et al. have introduced a new API Log_power_to_file, which can measure the power consumed by major components like CPU, Disk, RAM, GPU, and Xeon Phi [36].

3. Energy Consumption Analysis of NoSQL functional modules.

It is well said that if you have a hammer, you tend to see every problem as a nail. If one is planning to

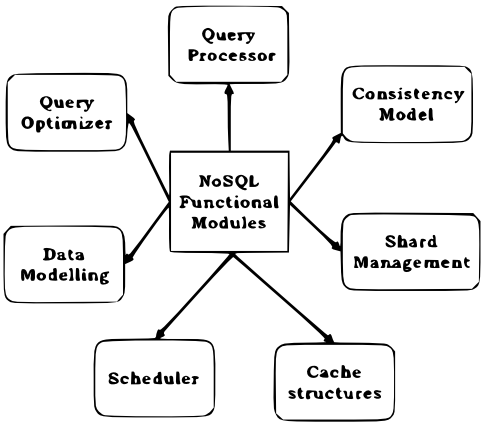

Fig. 3.1: Basic functional modules of NoSQLs

Table 3.1: Energy Consumption Analysis of NoSQL functional modules

Reference	Query optimization	Query Processing	Consistency	WEC	Latency	Data Modelling	Cloud pattern
[6]							✓
[14]	✓	✓					
[17]			✓				
[18]		✓					
[34]				✓			
[36]	✓						
[37]	✓						
[52]		✓					
[53]		✓				✓	
[54]		✓			✓		
[55]		✓			✓		
[57]					✓		

optimize the energy consumption of NoSQL, he tends to analyze the energy consumption of any functional module of NoSQL. Every NoSQL system has a different architecture but has a common subset of functional modules. Fig 3.1 shows the essential standard modules of NoSQL, where Data Modeling is the user interface module, and the rest are functional modules. This section presents the energy consumption analysis of various NoSQL modules. Despite being considered as a 'green system', NoSQL databases still lack mature solutions to evaluate and reduce Energy Consumption [34]. Only a few studies have analyzed the energy consumption of NoSQL. Table 3.1 presents a summary of our literature survey on energy consumption analysis of various functional modules of NoSQL databases.

There are more than 225 NoSQL databases introduced till now [45]. The webpage describes data model category-wise NoSQL databases. Major categories of NoSQL databases include Key-Value store, Wide column store, Document store, and Graph-based database. Every NoSQL has different architecture and functional modules. But, EC analysis is done on a limited set of NoSQL systems.

Table 3.2 shows a list of NoSQL for which energy consumption analysis is found in our literature survey. MongoDB is an example of a document store, Cassandra and HBase are column-store, Radis and Memcached are Key-Value stores, and Neo4j is a graph-based NoSQL. MongoDB and Cassandra are popular NoSQL for energy consumption analysis. These days many NoSQL databases are in-memory databases. So, this article also discusses the Energy consumption study of in-memory databases. The majority of the database operations

Table 3.2: Energy Consumption Analysis of NoSQL databases

Reference	MongoDB	Cassandra	HBase	Hive	Neo4j	Redis	Memcached
[6]	✓						
[14]	✓						
[17]			✓				
[18]	✓	✓				✓	
[34]		✓	✓	✓			
[36]	✓	✓					
[37]	✓	✓					
[52]					✓		
[53]	✓					✓	
[54]		✓					
[55]		✓					
[57]							✓

Table 3.3: Energy Consumption Analysis for Query operations

Reference	NoSQL	A	J	S	I	U	D	R	P	MR
[14]	MongoDB	✓	✓		✓					✓
[18]	MongoDB				✓	✓				
	Cassandra				✓	✓				
	Redis				✓	✓				
[34]	HBase	✓	✓		✓			✓	✓	
	Hive	✓	✓		✓			✓	✓	
	Cassandra	✓	✓		✓			✓	✓	
[37, 36]	Cassandra				✓	✓	✓			
	MongoDB				✓	✓	✓			
[52]	Neo4j	✓	✓	✓						
[53]	MongoDB	✓	✓		✓			✓		

A:Aggregate, J:Join, S:Sort, I:Insert, U:Update, D:Delete,
R:Range query, P:Pattern matching, MR:Map-Reduce

are memory-bound and waste computation power [44]. The impact of power management during various functions of the in-memory database on energy efficiency is presented in [28, 44], which is also included in this paper. Detailed energy consumption analysis of various NoSQL functional modules and resource management by in-memory databases are discussed in sub-sections.

3.1. Energy consumption analysis for Query Processing. Query processing is the heart of all database systems. It plays a key role in data analytics required for scientific or business intelligence. Therefore, the majority of energy consumption analysis of NoSQL systems is around query processing, which is presented in Table 3.3. It is infeasible to analyze the processing of each query of different applications. Instead, analysis of the type of queries and query operations would be a better choice. Although, at the initial stage of NoSQL Energy consumption analysis, researchers have started analyzing EC of a list of queries. Analysis of the energy consumption during query processing is presented at [53, 52, 14] for their query list, where [14] monitors EC of queries using aggregate and same queries using map-reduce operations. Similarly, the work represented at [37, 36] shows EC during query processing of some queries with an alternate query. The Energy consumption of Insert, Read, and Update operations for the increasing workload is presented at [18, 55]. Data access with patter analyzes energy consumption of Insert, Read, and Update operations. The energy consumption of Operations required for query access(Grep, Select), and for analytical access(Aggregate, Join) have been compared for different NoSQLs at [34].

Table 3.4: Expensive Query operations in terms of energy

Reference	MongoDB	Cassandra	Redis	Neo4j	HBase	Hive
[14]	Map-Reduce					
[18]	Insert, Update	Read	Insert			
[34]		Aggregate, Join, Grep			Aggregate, Join, Range Query, Reduce side Join	Aggregate, Join
[37, 36]	Update, Aggregate	Insert, Update, Aggregate, Search				
[52]				Join, Aggregate		

This paper has mapped query operations with test queries used in experiments conducted at various research papers. Table 3.3 describes a list of query operations analyzed and the corresponding NoSQL used for analysis. This summary helps researchers to identify further research scope in energy-efficient query processing. It shows the popularity of Insert, Aggregate, and Join operations. MongoDB is one of the widely used NoSQL, which is ensured from usage-based database ranking [24]. It might be an attraction point for more researchers to analyze energy consumption on MongoDB queries. Aggregate, Join, Insert, and Range queries are most popular. There are two groups of the survey found in this domain. One group has compared the impact of query operations on energy consumption by selected NoSQL with a relational database. At the same time, other groups have compared EC of query operations among various NoSQLs only. Both aspects will help find the scope of optimizing query processing of NoSQL in the context of energy consumption. Other perspectives may include a selection of the most energy-efficient NoSQL matching the application need and choosing alternate operations to get the query to execute.

Neo4j(NoSQL) consumes more energy than a relational database, especially for aggregate and join operations [52]. The work presented in [37, 36] reports the impact of query operation on energy consumption for simple dataset YCSB and complex dataset of Twitter. Both NoSQL (Cassandra and MongoDB) are found energy economical in comparison to MySQL for all test queries on YCSB. In contrast, their experiment result on Twitter data varies for different query types. For Insert and Aggregate query on Twitter data, MySQL consumes less energy consumption than both Cassandra and MongoDB. For Update, Delete, and simple Search queries on Twitter data, MongoDB consumes less energy than MySQL. Cassandra is found expensive in terms of energy spent for all types of queries except update queries on Twitter. The work also compares energy consumptions by query processing among different NoSQL databases (MongoDB and Cassandra). The result reveals that MongoDB is economical compared to Cassandra in the context of energy consumption by query processing.

The energy consumption by basic query commands (Insert, Read, Delete) on a different category of NoSQL have been compared in [18]. Cassandra is chosen as a column-oriented data store, MongoDB is taken as a document-oriented data store, and Redis is taken as Key/Value pair. Their result shows Cassandra consumes much more energy for the read operation, while Redis is more expensive at Insert queries. MongoDB consumes more power for the workload (1000 to 10000 operations) of Insert and Update than Redis and Cassandra. With increasing workload, energy consumption by MongoDB is reduced. In MongoDB, aggregate pipeline usage is found economic in both terms of energy and performance in comparison to map-reduce operations [14]. Table 3.4 summarizes the energy consumption analysis of query operations. It highlights query operations for each testbed NoSQL that consume more energy.

3.2. Energy consumption analysis for Query Optimization. Query Optimization is a pivotal component for any efficient database design. A huge amount of queries are executed daily. So, optimizing each

Table 3.5: Greenup Scenarios of optimization

Category	GreenUp	PowerUp	SpeedUp
1	> 1	1 (Power saving)	1 (Improve performance)
2	> 1	$= 1$ (No change in Power)	1 (Improve performance)
3	> 1	1 (Increase Power)	1 (Improve performance)
4	> 1	1 (Power saving)	1 (Degrade Performance)

query a little may help to improve the throughput and energy efficiency of a system respectively to a great extent. Unlike other software, a database system is the only software that can explore alternative ways of executing the query. The cost can also be query response time, latency, and energy consumption. Unlike NoSQL, a lot of research is done on optimizing queries to improve the energy efficiency of relational databases. In our information, [4] was the first to propose energy-aware query optimization. Later much other work on energy-aware Query Optimization was discussed for relational databases. [62, 61, 60, 58, 64, 22, 43, 20, 23].

Despite the continuous increasing usage of NoSQL databases, only [37] and its extended article [36] are found in our literature review that studies the impact of query optimization on the energy efficiency of NoSQL. The work depicted in [37, 36] illustrates some well-known basic query optimizations techniques for Cassandra and MongoDB. It also analyzes the impact of query optimization on power, performance, and energy efficiency. They believe that result of query optimization can be any of three or a combination of them: improve performance (Speedup >1), Power saving (PowerUp < 1), or improve energy efficiency (GreepUp > 1). They have tried to analyze four scenarios of GreenUp (Energy efficiency) as described in Table 3.5.

Category 1 is an ideal scenario, where improvement in both performance and power consumption results in energy saving. On another side, due to lack of energy consumption awareness, the 4th category is rarely observed. For MongoDB, their results say that using index fields on predicate for delete and covered query and project phase in aggregate query can help to achieve speedup with power saving. In contrast, sharding on multiple servers can improve performance at the cost of higher power consumption. MongoDB provides options to perform the write operation in bulk, and it can implement the bulk Insert operation either ordered or unordered. The general understanding is that the system implements unordered bulk write in parallel fashion and the ordered write operation in serial. Surprisingly counter-intuitive results that unordered write degrades performance and energy efficiency.

For Cassandra, row caching, LCS for read-heavy queries, and STCS optimizations for Insert-heavy queries also improved both performance and energy efficiency. They conclude that energy efficiency can be improved significantly on both MongoDB and Cassandra without degrading performance, but the improvement rate of energy efficiency is not linearly proportional to the speed of performance improvement. There are also scenarios where query optimization techniques may not be helpful for either performance improvement or energy efficiency improvement. Finally, they conclude three points: i) Query optimization can achieve energy efficiency for Cassandra and MongoDB without compromising performance. ii) Energy optimization is not always linearly proportional to performance optimization. And iii) query optimization technique may not optimize performance and decrease power all the time.

There is no other work in the literature that explicitly studies the impact of query optimization on energy consumption. Some work compares the energy consumption of queries using alternate ways of processing. Its result analysis can also help to redesign query optimization algorithms of NoSQLs. The work represented in [53], compares the energy consumption of sample queries on MongoDB with and without index. They conclude that the use of indexes helps to reduce energy consumption in most cases. Their other conclusion is application-level joins consume less energy than NoSQL-level joins. The work depicted in [14] describes energy consumption analysis for Insert and TPC-H (1,5,10,15,20) queries with and without index, with map-reduce, and with aggregate pipeline functions on MongoDB document store. Their result shows that the use of the Aggregate pipeline is more effective than the complex map-reduce process, and the use of the index is more effective for most queries with few exceptions like TPC-H query 1.

3.3. Energy consumption analysis of Consistency levels. None other than [17] article from our literature survey analyzing the impact of strong and eventual consistency on energy consumption and concurrency. The experiment comprises of 3 workloads(1. Write intensive(80% write), Read intensive(80% read), and mixed (50%read 50%write) that fully stress memory and exercise hard-disks on a columnar store HBase by applying the semantics of YCSB benchmark. They simulated these three types of workload over two configurations: i) Buffer based to simulate eventual consistency, which is the default configuration in HBase ii) Without Buffer to simulate eventual consistency.

The result reveals that strong consistency costs more in terms of energy on write-intensive workload, and eventual consistency costs more for the read-intensive and mixed workload. Finally, they think that change of request patterns by avoiding requests to unused disks and using caching to save energy consumption.

3.4. Waiting Energy Consumption Analysis . There exists much work analyzing the energy consumption of different software. A novel approach of reducing WEC to reduce energy wastage of NoSQL is proposed in [34]. WEC is one of the factors causing energy wastage due to computer idleness. It defines WEC as the energy wasted when some nodes are in a "passive idle" or "busy idle" state due to waiting for other resources.

The work chose four NoSQL databases (HBase, Cassandra, HadoopDB, and Hive), five types of queries (loading, and 4 query operations like fuzzy search, range search, aggregate, and join) to analyze WEC. The experiment result shows that NoSQL databases using a non-relational data model (like HBase and Cassandra) have high local and network I/O operations. NoSQL databases are I/O intensive, and the performance of the CPU is much higher than I/O operations done by the disk and network card. Hence, the CPU needs to wait for longer and more waiting energy consumption is produced. The paper describes that many NoSQL databases use the Map-Reduce model for query operations (like Selection, Aggregation, and Fuzzy Selection) and explicit use of Map Reduce. The result of the paper reveals that inappropriate Map Reduce can cause poor parallelism and poor synchronization, which generate waiting energy reduction.

One of the solutions to reduce WEC is to shut off idle systems to reduce energy wastage by idle nodes of a database cluster. But, this solution may not work for a NoSQL-like distributed system. Nodes cannot be shut off when they are temporarily idle but waiting for job scheduling, I/O operations, or computational results from other nodes. Hence, they suggest reducing waiting for energy consumption by lowering network I/O, synchronizing CPU and I/O operations, and proper Map-reduce framework selection considering data features.

3.5. Energy consumption analysis of Latency levels. Increasing reliance on the cloud has led to scale-out workloads, which are latency-sensitive. NoSQL servers need massive infrastructures to satisfy latency constraints, which consume more energy [6, 55, 57]. The work represented in [6, 55] describes the power consumption of the Cassandra cluster at 95th and 99th percentile latency for the read-only and write-only workload. Their result shows that read-only workloads need more energy consumption to maintain the 99th percentile compared to the 95th percentile latency, while update-only workloads do not need more energy to satisfy 95th to 99th percentile latency. They have also shown the impact of power provisioning and resource provisioning on energy saving, and their results say that resource provisioning can save more energy than power provisioning.

3.6. Energy consumption analysis of Data Modeling. Data Model provides a database user with a conceptual framework in which developers can specify the database requirements and structure to satisfy these requirements. The Document store NoSQL is one of the most popular NoSQL. It provides flexibility to choose data structures to represent data and their relationships. Parent Embedding, Child Embedding, Parent Referencing, Child Referencing, Two-way Embedding, Two-way Referencing, Bucketing, and De-normalization are primitive data models to specify any database. Data modeling influences query performance, consistency, and maintainability. Despite that, no work analyzes the impact of data modeling and schema design on energy consumption to our best knowledge. Only one work [53] has initiated investigating the effect of adopting these data models to design schema and energy consumption by executing queries on these schemas. Their result conveys some messages: i) There is no schema uniformly performing best for all queries, ii) No schema nor data models are consuming less energy for all queries. They suggest choosing a suitable data model based on required queries.

3.7. Energy consumption analysis of Cloud patterns for Database. With the increasing trend of Internet and Cloud computing, the inclination of many companies is toward cloud-based applications. Relational databases and NoSQL are two well-known database families used as the backbone of these cloud-based applications. Developers prefer to use cloud patterns to configure database systems to benefit from best practices [16]. Despite the wide adaptability of cloud patterns, only one work found in the literature studies the energy consumption of NoSQL while adopting cloud patterns. Therefore, this section discusses analysis done to study the impact of cloud patterns of NoSQL database systems on energy consumption.

The work depicted in [6] presents the impact of energy consumption of three cloud patterns [16]: Local Sharding Based Router, Local Database Proxy, and Priority Message Queue, with three databases: two popular relational databases (PostgreSQL and MySQL), and one NoSQL (MongoDB). In Local Database proxy, data is replication among a master node and the slave nodes, a proxy component route read requests and write requests to appropriate master and slave nodes. NoSQL database system is an excellent example of a distributed system. The majority of the NoSQL database uses the master/slave replication model, where every data chunk has one master copy, and other copies spread to other nodes are known as a slave. When the client requests for reading/write operations, the proxy's responsibility is to route all write operations to master and read procedures to slaves. With increasing data volume, NoSQL facilitates by splitting the database into multiple databases called shards. There are two well-known methods of sharding applied on shard keys: range-based sharding and hashing-based sharding. The local router routes a request to access the data in the Local Sharding-Based router. Priority Message Queue pattern is known for allowing asynchronous communication between components. It helps to improve the scalability of applications by supporting loosely coupled design. Priority Message Queue generally deals with different types of messages. They report the contrasting result. MongoDB executes faster than MySQL and consumes more energy than MySQL in cloud-based applications designed without adopting cloud patterns. Their results show that different cloud patterns impact relational and NoSQL database systems. Local Database Proxy improves the significant energy efficiency of MySQL while consuming more energy consumption on MongoDB. In adopting the Local Sharding Based Router pattern, the Modulo strategy strongly affects MongoDB, but a small for MySQL. Consistent and Lookup Strategy of Local Sharding Based router.

4. Proportionality and Trade-off. Most research works aim to make execution more and more fast. The fastest response is never the target of any application. Instead, desired performance expectations are specified using response time, latency, or throughput. On another side, Energy conservation is one of the significant research focuses these days. Energy consumption E of a NoSQL cluster of N nodes for the T period can be defined as shown in Eq. 4.1, where $P_i(t)$ is the power consumption of node i at time t.

$$E\left(T\right) = \sum_{i=1}^{N} \int_{0}^{T} P_i\left(t\right) dt \tag{4.1}$$

$$E\left(T\right) = P_{avg}\ X\ T \tag{4.2}$$

The cloud pattern analysis work depicted in [27] describes some essential points: i) Lookup and consistent hashing can improve energy consumption and performance. ii) Modulo algorithm does not improve performance or energy efficiency. iii) The pattern 'Local Database Proxy' can significantly improve the energy efficiency of cloud applications, while the pattern 'Local Sharding- based Router' combined with 'Local Database Proxy' can also improve response time without compromising energy efficiency.

There are two conflicting views showing the relation between energy and performance. One class of researchers believe that energy optimization comes byproduct of performance optimization, and another type of researcher believes that optimizing energy and performance are two conflicting targets. The first case can be verified by analyzing power consumption proportionality with performance measured. If energy is proportional to performance, there are two options to reduce energy consumption: i) Energy optimization as a result of performance optimization e.g. The application should take shortest execution time to achieve highest energy efficiency [11], ii) Select alternate operations that consume less power. Otherwise, a trade-off decision is required. Therefore, this section discusses the energy-performance proportionality study and trade-off analysis on NoSQL systems.

Table 4.1: Summary of Survey on Trade-offs in NoSQL

Reference	NoSQL	Metrics Analyzed for Trade-off
[37]	MongoDB, Cassandra	Performance (Query Response Time) and Energy Consumption
[36]	MongoDB, Cassandra	Performance(Query Response Time) and Energy Consumption
[55, 54]	Cassandra	Performance(Latency) and Energy Consumption
[17]	HBase	Consistency Level and Energy Consumption

4.1. Proportionality. Energy efficiency and Energy proportionality are major concerns today. As a result of researchers' effort, idle power consumption of database servers is reduced from 50% (in 2010) to 20% (today) of peak power consumption, which shows the trend of energy proportionality [28]. The energy-workload proportionality of Cassandra cluster for read-only workload and update-only workload is presented in [55, 54]. They exhibit poor energy proportionality in both workload types for all components except processor - CPU. CPU component is more energy proportional in read-only workload than update-only workload. The power consumption range for CPU and processor packages is from 30-100% and 55-100% in read-only workload and 78-100% and 82-100% in an update-only workload.

Dynamic power provisioning and resource provisioning are well-known techniques to control power consumption and resource allocation when the system is underloaded. The work also exhibits the impact of power provisioning and resource provisioning on the energy proportionality of the Cassandra cluster, where resource provisioning is much more effective. They have proposed hybrid provisioning (power provisioning and resource provisioning) technique to improve the energy proportionality to the next level.

Heterogeneous consumption of disks and memory instability usually causes power dis-proportionality of storage systems [17]. A comparison of energy consumption and execution time of various query operations is described in [14, 34]. It has observed almost the exact relationship between energy consumption and execution time. For example, HBase and Cassandra consume more energy as well as more execution time than Hive and HadoopDB for Loading, Grep, Selection, Aggregate, and Join operations [34]. Map-Reduce operation consumes more energy as well as execution time than Aggregation [14].

4.2. Trade-off. This section provide comprehensive survey on Trade-off in NoSQL databases. Fig. 4.1 summarize summary of Trade-off analysis done over different NoSQLs. Energy consumption cannot be considered independently of performance delivered by the system as they directly relate to each other [34, 48]. One view is that high performance costs high energy. So, it may be required to compromise performance for reducing energy consumption. Developing software techniques to achieve energy and performance trade-offs is one of the current research trend [8]. SLOs may specify performance requirements to decide the trade-off between performance and energy.

Most database researchers believe that a trade-off between power and performance is inevitable. There is a belief that many scenarios need to be analyzed for NoSQLs to understand the trade-off between energy optimization and performance optimization. The work shown in [36] is only work in our opinion analyzing the trade-off requirement between energy optimization and query optimization for the NoSQL database. They demonstrated a few well-known query optimization approaches on NoSQL databases (Cassandra and MongoDB) to analyze the impact of query optimization on energy optimization and performance optimization. For this, they have evaluated Powerup, SpeedUp, and GreenUp. If all these three conditions satisfy every time, energy optimization comes along with performance optimization. Conversely, if GreenUp and SpeedUp conditions are not satisfied for all scenarios without degrading PowerUp, then the performance optimization and energy optimizations seem two different goals. They showed that energy optimization is neither byproduct nor a conflicting goal of performance optimization in some situations and concluded that energy efficiency does not always scale linearly with performance. Hence, it shows research scope to analyze in detail trade-offs.

Latency SLOs are common these days. Performance targets (Latency SLOs) are typically based on 99th or 95th percentile in place average latency. This type of SLOs provides us an opportunity to trade performance

for power [55]. They have also demonstrated that compromising latency from 99th%-ile to 95th%-ile can also reduce the energy consumption of the Cassandra cluster. This work shows research avenues to make energy proportionality systems as wastage of energy cannot be ignored today, where the use of the energy-hungry technical device is continuously increasing.

Trade-offs between consistency level and energy for HBase NoSQL are analyzed in [17]. Strong consistency has better throughput-energy proportionality for all types of workload. On another side, eventual consistency shows better throughput-energy proportionality for the read-intensive and balanced workload. But, it is not proportional at under-loaded (low throughput). It also offers a trade-off scenario at write workload, where strong(high) consistency results at the cost of spending more energy.

5. Energy Conservation in NoSQL. Reducing the energy consumption of NoSQL or improving the energy efficiency of NoSQL requires proper knowledge of energy consumption by NoSQL. section 3 describes a little effort on the energy consumption analysis of NoSQL. During the literature survey, only four papers were found that propose techniques to reduce the energy consumption of NoSQL. All these papers touch on different functional modules of NoSQL. It includes optimizing energy proportionality [55], the trade-off between memory performance and power [57], low-power database server for IoT [47], and auto-scaling of virtual data centers [9].

The total system energy of the Cassandra cluster is poorly proportional to workload, especially when the system is underloaded [55]. The work presented in [55] investigates the effect of power management techniques on energy proportionality, where resource provisioning results better compared to power provisioning techniques. To improve energy proportionality to the next level, they have proposed hybrid (resource + power) provisioning and trade latency by considering the difference between measured latency and SLO. The proposed hybrid provisioning with 95%-ile latency delivers the most power-saving (up to 55%.)

A hardware-software unified solution offers a trade-off memory performance with power. Lake [57] : a Low Latency, power-efficient Key-value store design to improve power efficiency. They present a multi-level multi-core cache design after exploring the trade-off between performance and power by leveraging different types of on-chip and onboard memories. It claims low latency (1.1μs on hit) and better throughput (13.1Mqps) at the cost of 10W additional power.

The work depicted in [47] proposes a methodology to construct a low-power database server for IoT middleware. The work uses Raspberry Pi, and MongoDB. It presents two designs: i) Non-data Copy Oriented method: The client sends a request to the master node. The Master node finds a data node matching the client request and sends it an activating signal. Then, the client node will send data to the master node. ii) Data copy oriented from data node to the master, where each data node periodically transfers data to the master node. Here, the master node should have ample storage.

6. Discussion and Future Directions. NoSQL systems are used widely for the development of web applications, data analytics, and IoT systems. NoSQLs have started taking place of RDBMSs to serve better scalability, availability, performance, and variety of data handling. This section describes gaps or limitations observed in related work done, future research directions, and challenges to do energy consumption research on NoSQL.

Benchmark consumption is the biggest issue for research in NoSQL. Despite the wide use of NoSQL, researchers could not find a suitable benchmark for NoSQL except YCSB [41]. YCSB framework provides a set of test cases combined by Insert, Read, Update and Scan operations [36], which could be adopted to measure the performance of NoSQL databases [10]. Still, these test cases only involve too simple database operations [34].

Table 6.1 describes benchmarks used to test the energy consumption of NoSQL. Most research on NoSQL uses YCSB and TPC benchmarks. TPC benchmarks are designed to test relational databases' performance and are far away from NoSQL databases. For example, i) TPC-benchmark supports ACID, and NoSQL supports the BASE. ii) TPC-H contains complicated queries containing Join, Group, and Aggregate operations. While, many NoSQLs do not support explicit interfaces for Join, Group-by, and Aggregate operations. SSB is available to test the performance of the data warehouse, and SSB design is based on TPC-H only. Looking toward the distinct feature of NoSQL and relational databases, it is not preferred to apply any benchmark of the relational database to non-relational databases. In addition, every NoSQL also has different characteristics. Another problem is that most benchmarks available for databases are to test performance. TPC-Energy is a benchmark for a relational database, which allows examining the energy consumption of the servers [27]. Hence, it is

Table 6.1: Benchmarks

	YCSB	Twitter	TPC-H	Other
[6]				DVD store [15], JPteSTore [30]
[9]	✓			
[14]			✓	
[17]	✓			
[18]	✓			
[28]				TATP(OLTP) [21], SSB(OLAP) [46]
[34]				Own
[36]	✓	✓		
[37]	✓			
[52]				Adventure Data Warehouse [38]
[54]				Cloudsuit [50]
[55]	✓			

required to design benchmarks to test NoSQL servers' performance and energy consumption. Table 6.2 briefly describes other gaps or limitations observed in the literature survey and shows some future directions.

7. Conclusion. Despite the wide use of NoSQL and knowing NoSQL consumes more energy consumption, NoSQL is lagging for energy conserving optimization. Therefore, this article presents a comprehensive survey on energy consumption analysis for NoSQL databases. This paper classifies the analysis work as per NoSQL functional modules. This work collects results from different papers and generates various summaries in tables. It includes components consuming significant power, a list of NoSQL analyzed, a list of NoSQL functional modules analyzed for its EC, a list of query operations monitored, and query operations consuming high energy. Little effort is made to reduce the energy consumption of NoSQL, which is also presented here. Finally, the paper discusses gaps in the articles surveyed, summarizes recommendations from the articles, and directs future research scope toward conserving energy consumption of NoSQL systems.

REFERENCES

[1] *Cisco Global Cloud Index: Forecast and Methodology, 2016–2021 White Paper.* `https://techie.buzz/2019/03/19/cisco-global-cloud-index-forecast-and-methodology-2016-2021-white-paper/`, 2019. Accessed:2019-03-29.

[2] *Work, energy, and energy resources.* `https://opentextbc.ca/openstaxcollegephysics/chapter/world-energy-use/`. Accessed:2022-03-29.

[3] R. Agrawal, A. Ailamaki, P. A. Bernstein, E. A. Brewer, M. J. Carey, S. Chaudhuri, A. Doan, D. Florescu, M. J. Franklin, H. Garcia-Molina, et al., *The claremont report on database research*, ACM Sigmod Record, 37 (2008), pp. 9–19.

[4] R. Alonso and S. Ganguly, *Energy efficient query optimization*, in Matsushita Info Tech Lab, Citeseer, 1992.

[5] B. Bani, *Understanding the impact of databases on the energy efficiency of cloud applications*, PhD thesis, Ecole Polytechnique, Montreal (Canada), 2016. Thesis.

[6] B. Bani, F. Khomh, and Y.-G. Guéhéneuc, *A study of the energy consumption of databases and cloud patterns*, in International Conference on Service-Oriented Computing, Springer, 2016, pp. 606–614.

[7] R. Bolla, R. Bruschi, and P. Lago, *The hidden cost of network low power idle*, in 2013 IEEE International Conference on Communications (ICC), IEEE, 2013, pp. 4148–4153.

[8] B. Bylina, J. Potiopa, M. Klisowski, and J. Bylina, *The impact of vectorization and parallelization of the slope algorithm on performance and energy efficiency on multi-core architecture*, in 2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS), IEEE, 2021, pp. 283–290.

[9] E. Casalicchio, L. Lundberg, and S. Shirinbab, *Energy-aware auto-scaling algorithms for cassandra virtual data centers*, Cluster Computing, 20 (2017), pp. 2065–2082.

[10] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, *Benchmarking cloud serving systems with ycsb*, in Proceedings of the 1st ACM symposium on Cloud computing, 2010, pp. 143–154.

[11] D. Davidović, M. Depolli, T. Lipić, K. Skala, and R. Trobec, *Energy efficiency of parallel multicore programs*, Scalable Computing: Practice and Experience, 16(4) (2015), pp. 437–448.

[12] M. Dayarathna, Y. Wen, and R. Fan, *Data center energy consumption modeling: A survey*, IEEE Communications Surveys & Tutorials, 18 (2015), pp. 732–794.

Table 6.2: Gap Analysis and Future Direction

Key element	Gap analysis and future directions
RDBMS and NoSQL	• Energy and Performance difference analysis between NoSQL and RDBMS is available for a limited category of NoSQL only. Researchers can explore similar research for different types of NoSQL(like Document Oriented, Graph-based, Key-Value Pair, Column Family, etc.). • NoSQL systems consume more energy than relational databases [6, 52, 37, 36], where they also use similar deployment setup for both NoSQL and RDBMS. But, Which component or function of NoSQL consumes more energy and Why? are still in the research scope. • An empirical study in [6] describes that despite a similar setup (deployment to a distributed cluster) in RDBMS and MongoDB, query response time and energy consumption patterns conflict with each other. Therefore, It is suggested to explore a relationship between energy consumption and utilization of each component of each cluster node in active and idle states both.
Power monitoring	• Usually, it is a practice to use a distributed cluster setup for NoSQL deployment. But, section 2 does not show power consumption analysis of network resources nor specify a reason to ignore it. • Although the introduction to Waiting Energy Consumption in [34], a method of monitoring waiting energy is still unknown.
Power Wastage	• Energy-workload proportionality and WEC study seem promising techniques to identify power wastages. Power management is suggested for the underloaded situation [55]. A comprehensive survey on power management techniques are presented in [56]. • Traditional resource provisioning algorithms may not be directly applicable to NoSQL. Is it possible to design resource provisioning algorithm for a distributed NoSQL, which do not deteriorate other quality of service like scalability, availability, etc • Waiting energy consumption analysis presented in [34] recommends reducing CPU waiting time and WEC by optimizing data format, storage and selecting appropriate scheduling algorithms. • Analysis done to improve power efficiency and latency in [57] strongly suggest to reduce speed gap between network I/O and Computation.
Query Optimization	• The work presented in [37, 36] have explored the impact of query optimization on energy efficiency for very limited query operations and with a single alternate query plan of MongoDB and Cassandra. The work can be extended for various types of query operations along with promising plans.
Data Modeling	• There are many factors like query structure, data model, index usage, and query optimization in document stores that affect query performance [36]. Data modeling is a process to define and structure data elements in the context of the relevant application. Only one article draws our attention to data structuring [53]. Proposing data structures considering their impact on energy consumption will be a great contribution to the energy conservation of NoSQL. • [36] have observed the different impacts of query processing on different datasets, where datasets were models using two different data models i.e. Denormalized, Child Embedding. It shows need of analyzing energy consumption impact of different query patterns on possible set of data models.
Tools	• Like performance profiler, energy profiler tools can be designed to generate the energy consumption profile of a distributed NoSQL and highlight energy hotspot modules.
Energy Conservation	• Energy conservation of NoSQLs is still in future scope as proper energy consumption analysis is fundamental for it.

[13] A. d'Azemar, ldesauw, B. Jordan, G. Fieni, and Guillaume, *Powerapi-ng/rapl-formula*.
[14] D. Duarte and O. Belo, *Evaluating query energy consumption in document stores*, in International Conference on Emerging Technologies for Developing Countries, Springer, 2017, pp. 79–88.
[15] *Dell dvd store database test suite*. http://linux.dell.com/dvdstore/, 2011. Accessed:2022-03-29.
[16] C. Fehling, F. Leymann, R. Retter, D. Schumm, and W. Schupeck, *An architectural pattern language of cloud-based applications*, in Proceedings of the 18th Conference on Pattern Languages of Programs, 2011, pp. 1–11.
[17] Á. García-Recuero, *On the energy efficiency of client-centric data consistency management under random read/write access to big data with apache hbase*, arXiv preprint arXiv:1509.02640, (2015).
[18] C. Gomes, E. Tavares, and M. N. d. O. Junior, *Energy consumption evaluation of nosql dbmss*, in Anais do XV Workshop em Desempenho de Sistemas Computacionais e de Comunicação, SBC, 2016, pp. 71–81.
[19] R. Gonçalves, J. Saraiva, and O. Belo, *Defining energy consumption plans for data querying processes*, in 2014 IEEE Fourth International Conference on Big Data and Cloud Computing, IEEE, 2014, pp. 641–647.
[20] G. Graefe, *Database servers tailored to improve energy efficiency*, in Proceedings of the 2008 EDBT workshop on Software engineering for tailor-made data management, 2008, pp. 24–28.
[21] I. S. Group, *Ibm software group information management, telecom application transaction processing (tatp) benchmark description*. http://tatpbenchmark.sourceforge.net/TATP_Description.pdf. Modified Date: 2009-03-27 Accessed:2022-03-29.
[22] B. Guo, J. Yu, B. Liao, D. Yang, and L. Lu, *A green framework for dbms based on energy-aware query optimization and energy-efficient query processing*, Journal of Network and Computer Applications, 84 (2017), pp. 118–130.
[23] S. Harizopoulos, M. Shah, J. Meza, and P. Ranganathan, *Energy efficiency: The new holy grail of data management systems research*, arXiv preprint arXiv:0909.1784, (2009).
[24] S. IT, *Db-engines ranking*. https://db-engines.com/en/ranking. Accessed:2022-03-29.
[25] K. N. Khan, M. Hirki, T. Niemi, J. K. Nurminen, and Z. Ou, *Rapl in action: Experiences in using rapl for power measurements*, ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS), 3 (2018), pp. 1–26.
[26] B. Khargharia, S. Hariri, and M. S. Yousif, *Autonomic power and performance management for computing systems*, Cluster computing, 11 (2008), pp. 167–181.
[27] F. Khomh and S. A. Abtahizadeh, *Understanding the impact of cloud patterns on performance and energy consumption*, Journal of Systems and Software, 141 (2018), pp. 151–170.
[28] T. Kissinger, D. Habich, and W. Lehner, *Adaptive energy-control for in-memory database systems*, in Proceedings of the 2018 International Conference on Management of Data, 2018, pp. 351–364.
[29] M. Kunjir, P. K. Birwa, and J. R. Haritsa, *Peak power plays in database engines*, in Proceedings of the 15th International Conference on Extending Database Technology, 2012, pp. 444–455.
[30] J. Landis, K. Shimizu, E. Macarron, I. AVE, G. Gotimer, R. Bala, I. Baiborodine, K. Zero, H. Boutemy, and S. Tripodi, *Mybatis jpetstore*. https://github.com/mybatis/jpetstore-6. Accessed:2022-03-29.
[31] W. Lang, R. Kandhan, and J. M. Patel, *Rethinking query processing for energy efficiency: Slowing down to win the race.*, IEEE Data Eng. Bull., 34 (2011), pp. 12–23.
[32] W. Lang and J. Patel, *Towards eco-friendly database management systems*, arXiv preprint arXiv:0909.1767, (2009).
[33] J. B. Leverich, *Future scaling of datacenter power-efficiency*, Stanford University, 2014.
[34] T. Li, G. Yu, X. Liu, and J. Song, *Analyzing the waiting energy consumption of nosql databases*, in 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, IEEE, 2014, pp. 277–282.
[35] Y. Li and S. Manoharan, *A performance comparison of sql and nosql databases*, in 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), IEEE, 2013, pp. 15–19.
[36] D. Mahajan, C. Blakeney, and Z. Zong, *Improving the energy efficiency of relational and nosql databases via query optimizations*, Sustainable Computing: Informatics and Systems, 22 (2019), pp. 120–133.
[37] D. Mahajan and Z. Zong, *Energy efficiency analysis of query optimizations on mongodb and cassandra*, in 2017 Eighth International Green and Sustainable Computing Conference (IGSC), IEEE, 2017, pp. 1–6.
[38] MashaMSFT, M. Ghanayem, JulieMSFT, D. Coulter, KatsutoshiOtogawa, J. Parente, M. Ray, T. Petersen, J. Wells, and J. Rytlewski, *Adventureworks sample databases*. https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15&tabs=ssms. Accessed:2022-03-29.
[39] F. Mehdipour, H. Noori, and B. Javadi, *Energy-efficient big data analytics in datacenters*, in Advances in Computers, vol. 100, Elsevier, 2016, pp. 59–101.
[40] L. Minas and B. Ellison, *Energy efficiency for information technology: How to reduce power consumption in servers and data centers*, Intel press, 2009.
[41] M. J. Mior, K. Salem, A. Aboulnaga, and R. Liu, *Nose: Schema design for nosql applications*, IEEE Transactions on Knowledge and Data Engineering, 29 (2017), pp. 2275–2289.
[42] S. Mittal, *Power management techniques for data centers: A survey*, arXiv preprint arXiv:1404.6681, (2014).
[43] R. Niemann, *Towards the prediction of the performance and energy efficiency of distributed data management systems*, in Companion Publication for ACM/SPEC on International Conference on Performance Engineering, 2016, pp. 23–28.
[44] S. Noll, H. Funke, and J. Teubner, *Energy efficiency in main-memory databases*, Datenbank-Spektrum, 17 (2017), pp. 223–232.
[45] *List of nosql database management systems*. https://hostingdata.co.uk/nosql-database/. Accessed:2022-03-29.
[46] P. O'Neil, B. O'Neil, and X. Chen, *Star schema benchmark*. https://www.cs.umb.edu/~poneil/StarSchemaB.PDF. Accessed:2022-03-29.
[47] P. Paethong, M. Sato, and M. Namiki, *Low-power distributed nosql database for iot middleware*, in 2016 Fifth ICT

international student project conference (ICT-ISPC), IEEE, 2016, pp. 158–161.
[48] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath, *Load balancing and unbalancing for power and performance in cluster-based systems*, (2001).
[49] M. Poess and R. O. Nambiar, *Energy cost, the key challenge of today's data centers: a power consumption analysis of tpc-c results*, Proceedings of the VLDB Endowment, 1 (2008), pp. 1229–1240.
[50] P. Pourhabibi, Javier, M. Otto, D. Ustiugov, ivonindza, C. Lin, and aledaglis, *Cloudsuite*. `https://github.com/parsa-epfl/cloudsuite`. Accessed:2022-03-29.
[51] N. Rasmussen, *Determining total cost of ownership for data center and network room infrastructure*, Relatório técnico, Schneider Electric, Paris, 8 (2011).
[52] J. Saraiva, M. Guimarales, and O. Belo, *An economic energy approach for queries on data centers*, (2017).
[53] M. Shah, A. Kothari, and S. Patel, *Influence of schema design in nosql document stores*, in Mobile Computing and Sustainable Informatics, Springer, 2022, pp. 435–452.
[54] B. Subramaniam and W. Feng, *On the energy proportionality of scale-out workloads*, arXiv preprint arXiv:1501.02729, (2015).
[55] B. Subramaniam and W.-c. Feng, *On the energy proportionality of distributed nosql data stores*, in International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems, Springer, 2014, pp. 264–274.
[56] A. Thakkar, K. Chaudhari, and M. Shah, *A comprehensive survey on energy-efficient power management techniques*, Procedia Computer Science, 167 (2020), pp. 1189–1199.
[57] Y. Tokusashi, H. Matsutani, and N. Zilberman, *Lake: An energy efficient, low latency, accelerated key-value store*, arXiv preprint arXiv:1805.11344, (2018).
[58] Y.-C. Tu, X. Wang, B. Zeng, and Z. Xu, *A system for energy-efficient data management*, ACM SIGMOD Record, 43 (2014), pp. 21–26.
[59] J. Wang, L. Feng, W. Xue, and Z. Song, *A survey on energy-efficient data management*, ACM SIGMOD Record, 40 (2011), pp. 17–23.
[60] Y. Xu, Zichen and Tu and X. Wang, *Dynamic energy estimation of query plans in database systems*, in 2013 IEEE 33rd International Conference on Distributed Computing Systems, IEEE, 2013, pp. 83–92.
[61] Z. Xu, Y. Tu, and X. Wang, *Pet: reducing database energy cost via query optimization*, Proceedings of the VLDB Endowment, 5 (2012), pp. 1954–1957.
[62] Z. Xu, Y.-C. Tu, and X. Wang, *Exploring power-performance tradeoffs in database systems*, in 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), IEEE, 2010, pp. 485–496.
[63] Z. Xu, X. Wang, and Y. Tu, *Power-aware throughput control for database management systems*, in 10th International Conference on Autonomic Computing ({ICAC} 13), 2013, pp. 315–324.
[64] Z. . Xu, Y. Tu, and X. Wang, *Online energy estimation of relational operations in database systems*, IEEE transactions on computers, 64 (2015), pp. 3223–3236.

Edited by: Fabrizio Marozzo
Received: Jan 21, 2021
Accepted: Apr 3, 2022

AIMS AND SCOPE

The area of scalable computing has matured and reached a point where new issues and trends require a professional forum. SCPE will provide this avenue by publishing original refereed papers that address the present as well as the future of parallel and distributed computing. The journal will focus on algorithm development, implementation and execution on real-world parallel architectures, and application of parallel and distributed computing to the solution of real-life problems. Of particular interest are:

Expressiveness:

- high level languages,
- object oriented techniques,
- compiler technology for parallel computing,
- implementation techniques and their efficiency.

System engineering:

- programming environments,
- debugging tools,
- software libraries.

Performance:

- performance measurement: metrics, evaluation, visualization,
- performance improvement: resource allocation and scheduling, I/O, network throughput.

Applications:

- database,
- control systems,
- embedded systems,
- fault tolerance,
- industrial and business,
- real-time,
- scientific computing,
- visualization.

Future:

- limitations of current approaches,
- engineering trends and their consequences,
- novel parallel architectures.

Taking into account the extremely rapid pace of changes in the field SCPE is committed to fast turnaround of papers and a short publication time of accepted papers.

INSTRUCTIONS FOR CONTRIBUTORS

Proposals of Special Issues should be submitted to the editor-in-chief.

The language of the journal is English. SCPE publishes three categories of papers: overview papers, research papers and short communications. Electronic submissions are preferred. Overview papers and short communications should be submitted to the editor-in-chief. Research papers should be submitted to the editor whose research interests match the subject of the paper most closely. The list of editors' research interests can be found at the journal WWW site (`http://www.scpe.org`). Each paper appropriate to the journal will be refereed by a minimum of two referees.

There is no a priori limit on the length of overview papers. Research papers should be limited to approximately 20 pages, while short communications should not exceed 5 pages. A 50–100 word abstract should be included.

Upon acceptance the authors will be asked to transfer copyright of the article to the publisher. The authors will be required to prepare the text in LaTeX 2$_\varepsilon$ using the journal document class file (based on the SIAM's `siamltex.clo` document class, available at the journal WWW site). Figures must be prepared in encapsulated PostScript and appropriately incorporated into the text. The bibliography should be formatted using the SIAM convention. Detailed instructions for the Authors are available on the SCPE WWW site at `http://www.scpe.org`.

Contributions are accepted for review on the understanding that the same work has not been published and that it is not being considered for publication elsewhere. Technical reports can be submitted. Substantially revised versions of papers published in not easily accessible conference proceedings can also be submitted. The editor-in-chief should be notified at the time of submission and the author is responsible for obtaining the necessary copyright releases for all copyrighted material.