SEARCH OPTIMISATION USING DATA ANALYTICS

Darshan R P (1BM14CS035), Hemanth P (1BM14CS121), Shankar R (1BM14CS133), Sushmitha S Jois (1BM14CS097)

SECTION 1.

ABSTRACT: ***Big data analytics involves usage of various techniques and methods for the extraction of useful information from data having large volume, velocity and variety. With the huge amounts of data that is being produced in today's world, big data analytics is the key to breaking down this massive amount of knowledge to find hidden patterns and gain insights from it. Searching this prodigious amount of data to find required details can be a laborious and time-consuming task. This paper proposes a fast approach for search based data retrieval. The aim is to achieve significantly low processing time and show the results by implementing it on an application for pet related searches. The algorithm used is aimed at being general so that it can be applied to various similar search applications.***

Here, we discuss the importance of big data analytics combined with the usage of cloud for storage and how it can enhance our searches in future. The tools and techniques are reviewed. Conclusions are drawn concerning the design and the applications of this project.

SECTION 2.

INDEX TERMS:

Big Data Analytics: Big data analytics (BDA) applications are defined as a new category of software applications that process large amounts of data using large scale parallel processing infrastructure to uncover hidden patterns and improve retrieval efficiency.

Cloud Computing: The practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer to enable access from various parts of the globe.

Efficient searches: The main objective here is to reduce the time required to access, retrieve and filter real time data to obtain the required results.

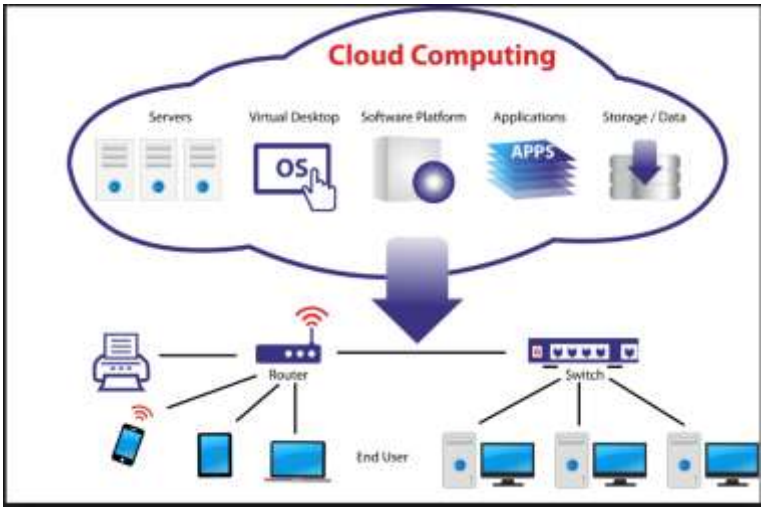
SECTION 3.
INTRODUCTION:

Search based applications are applications that have a search engine as their base and are good at slicing and dicing large datasets to give out only the required subset of data. For instance, eBay has deployed BDA applications to optimise product search by processing 5 PB data with more than 4000 CPU cores, and Facebook repeats collecting and analysing more than 700 TB data every day to drive its core business. The advances in technology these days are affecting lives in various aspects be it business, medical or educational fields. The most upcoming and intriguing amongst these in the world of computer science are that of cloud computing, big data analytics and data science and mobile applications. Big data analytics deals with data with large volume, velocity and variety and performs operations on this data such as parsing, pattern matching, searching and text mining to discover the various relations and patterns among the data. Big data analytics usually processed large amounts of data, 100 TB or more sometimes, and for this it requires high performance processors that provide the required results, efficiently.

This is where cloud computing plays a vital role as it provides huge storage spaces and also attends to the computational requirements. Based on the rate at which people are moving over to cloud based architecture when compared to storing data locally, big data analytics would eventually rely on cloud computing for efficient task completion. Data analytics along with cloud computing can help people take better decisions since they will be able to access larger amount of information, compare the various factors and accordingly take a well-informed decision.

For example, if a person wants to buy a car, it would be easier for him to do so if he could access data as to which car satisfies his criteria better. He can check the ratings various cars have been given and also compare the same. Also, if analysis is done as to which is the best one to buy based on his budget it would be very simple for him to take a decision. Data analytics will help people observe patterns and gain knowledge to take well informed and pragmatic decisions. Thus, big data analytics will help take better, well informed decisions and improve our choices in life.

Cloud computing has made the access and storage of data much safer and more efficient. Data can be stored and accessed from anywhere with ease. It is currently being used by various organizations and individuals to enhance productivity and performance but at the same time, reduce costs, complexity and investments. Cloud computing relies on a set of network-connected resources shared to maximize their utilization resulting in reduced management and capital costs. Cloud computing ensures that users do not have to worry about limited space or computing capacity and overcomes a few limitations of using mobile devices.



For example, given present advent in mobile technology, multimedia applications are found to be the most popular. But they also face certain challenges such as high computing capabilities, huge storage space, and more security. Cloud computing helps us to overcome these challenges by storing all the data on cloud and making it available for the users whenever they need. Of Course, with these advantages, there are a few limitations such as delay when the mobile user is accessing the data from cloud from remote places. And along with this storage issue, there is also a challenge that is faced when it comes to the privacy of a user's data. There are few other concerns like availability of data at any time and maintaining the integrity the data.

To obtain the required result efficiently, big data and other relevant technologies provide help in terms of data management and analytics. These technologies are designed to obtain the result from data that consists of the three Vs characteristic; volume, velocity, and variety. This data analysis helps us to find out all the hidden information and pattern, and if studied and recognised, decision making can be improved.

We wish to create an application that helps users find pets that are up for sale and the vets in and around their area by detecting their location. To accomplish this, we shall integrate mobile app development along with big data analytics which makes our job easier and also helps in the betterment of the business. A pet is always considered to be very close to humans and also people these days think pets play a major role in their lives. They go to great extent in order to own pets and take good care of them. But in the present market, there are no application that helps a user to buy pet online or put up pets for adoption. So, we want to create an application that will bring joy and pleasure to a user when it comes to their pets' life. This application will use big data analytics to help the users in their searches.

The main aim is to improve the efficiency of these searches and reduce the amount of time needed to obtain the required results. We will accomplish this by combining big data analytics with data science that helps to overcome the limitations mentioned above. The paper consists of the following sections: Section II presents the literature review, and Section III discusses the system we wish to innovate. Section IV presents big data analytics, followed by a review of data analytics tools in Section V. Section VI concludes the paper and provides an outlook for search optimisation techniques.

SECTION 4.
RELATED WORK:

A. Data Collection

Data collection is the process of gathering information from various sources in order to later analyse this data and gather insights from it. When the data involved is of high velocity, high variety and high volume, it is called big data. This large amount of data being generated every day is present across multiple websites. There are a number of automated techniques to access this data from websites, most of which are ad-hoc and domain specific. The architecture proposed by the authors of [1] offers an easy and feasible approach for parsing and extracting data on a large scale from multiple websites with minimal human intervention. [2] proposes a single cloud based architecture for scraping of data as well as managing the feasibility of big data applications. The scalability and performance of the proposed scrapper is analysed.

B. Searching

"A brief study and analysis of different Searching Algorithms" [3], discusses about the various searching algorithms that can be used to improve the efficiency of searches. Linear search is an algorithm that searches all elements of a data set to find the result or the required item. It is easy and resource efficient, but consumes a lot of time. Binary search is more efficient to search for a particular data from a sorted list. Interpolation search works by calculating the probing position in the list. Jump search algorithm uses minimum gapping value to search particular data.

A fast string searching algorithm was proposed by researchers Robert S Boyer et al. in the year 1977, which foresights the string being searched for. Here the matching is done from right to left rather than the conventional direction of left to right to find matches and patterns. If the last character of the searched character name is matched then it will examine the second last of the character name and so on until we get a positive result or terminating the process getting a negative output [4]. This method has a low execution value for best case but for worst case it takes much more time to execute the program.

The researchers Phisan Kaewprapha et al. proposed a search algorithm named Network localization using tree search in the year 2016, the algorithm is a heuristic process which works under the presence of anchor nodes (nodes, whose locations are well known) to find the locations of desired unknown nodes [5]. The addition of a new node in the existing list is done by traversing and checking it's compatibility for neighbouring nodes and non-neighbouring nodes. Apart from solving a problem faster this method does not promise to be optimal.

Researchers Kashale Chimmanga et al. proposed a search algorithm i.e. an application of best first search algorithm to demand control in the year 2016 to minimize the deficiency of electricity in Zombia. The algorithm searches for the best combination of household appliances according to their power rate and ranking. The algorithm runs through each and every appliance and alerts the user as to which appliance should be switched on or off as per their priorities. The priority (represented by a number) is given by the user. The main advantage of Best first search algorithm over Breadth first search algorithm is that as it uses heuristic functions. It is applicable for large number of appliances and produces result much faster [7]. But sometimes the result may be disvalued if the heuristic function so produced contains some errors.

A nearest neighbor search algorithm for LR-LD on high SNR was proposed by researchers Thae Thae Yu Khine et al. in the year 2016. This algorithm includes two parts that is derivation of unimodular matrix and feasible detection of symbols. But it mainly focuses on the second drawback of lattice reduction (LR) technique (feasible set of the detected symbol cannot be found without huge operation) [6]. It produces feasible set of detected symbols without lesser amount of procedures and calculations. The algorithm work same as of previous LR algorithm to derive the unimodular(matrix having perpendicular vectors) matrix[8].Its operation is easy and execution time is much lower.

"Opiner: An Opinion Search and Summarization Engine for APIs" [28] discusses about how opinions play an important role in activities related to software development. The perceptions about an API depends on how the developer will see and evaluate the API. It is very difficult to make an informed decision when there are so many APIs that are made available. The server side of Opiner collects and summarizes the opinion about an API and the client side uses a website to present the same. Opiner was evaluated for two development tasks and was found to help the developers make the right decision. APIs offers interfaces to reusable software components and are also a part of software development. Thus, with huge amounts of APIs available for various types of development task, it is very hard to select the right one. There are a lot of developer forums that help in communication to discuss and choose an API. In Opiner, when an input is given it gives you the summary of all reviews pertaining to an API. It also helps you to search for an API, see all reviews about the API, search of API aspect and also find API ratings. The developers were given access to Opiner and Stack Overflow. But when developers used only Stack Overflow there was around 66% accuracy but when they used both Stack Overflow and Opiner there was 100% accuracy. There were three challenges faced – API Mention Detection, API Opinion Association, API Opinion Summarization. Opiner architecture supported efficient implementation of algorithms and a web-based search engine with visual representation. There were four major components – Application Engine, Database, REST Web Server, and Website. An investigation took place as to how the developers were able to make decision when both Stack Overflow and Opiner were used. They detected API mentions using the technique [28] in Opiner for this evaluation. Opiner was the first tool to automatically mine and summarize opinions about an API.

"Generating Text Search Applications for Databases" [57] discusses how to connect a domain analysis tool's output to a program generator and to implement the generator. The real-world application here involves text searching analysis. There are works on domain analysis but however to obtain domain-based reuse, the outputs of the domain analysis phase and the inputs of the domain implementation phase have to be linked. Domain engineering is the process of creating an infrastructure to support systematic reuse. It has two phases: Domain Analysis which is identifying and documenting the commonalities and Domain Implementation which is to develop and implement reusable domain assets based on knowledge got from domain analysis. The first phase must produce a domain model. It includes the domain's scope, its vocabulary, and commonalities and variabilities. Methods of domain analysis include - the Domain Analysis and Reuse Environment [52], Family-Oriented Abstraction, Specification, and Translation [53], and Feature-Oriented Domain Analysis [54]. DARE is the domain analysis tool used here. Text search application is a subtopic in information retrieval. Users put across the information as queries and expect the required result. If not, they will modify the query and obtain the result. Text search application's architecture comprises five main components – Index creation and maintenance, Query parser, Search, User interface, and Document services. Commonalities are decisions made during the domain analysis phase about what is common across a wide range of search applications. Variabilities are decisions made during application generation or use about domain engineering. Using DARE-web, all assets are captured and stored. Based on system or component specification, a generator returns a finished system or component. This specification can be textual or interactive [55]. A compiler will then translate the output to an executable, once the generator is done. Research in the area of application generators has always been active [56]. XML is used as a specific language because it is used in wide range of application and adopted my many companies as well because it is human –readable, machine-understandable, and has general syntax. The generator can work with the DARE-Web implementation, with a user interface, or from the command line as a standalone component. The high-level product information and the configuration data are separated because there are a lot of information about new features and old features. The overall research gave confidence to productize the application generator. A wide range of Oracle developers have used the wizard for different tasks. This project shows that we can use Oracle to build an infrastructure that supports domain engineering at a low cost of ownership, hook DARE-Web to a code generation process, thus achieving systematic reuse at the requirements level, link a domain analysis tool such as DARE to a code generator using XML tools, create successful database applications with XML, even though as a specification language it has disadvantages over other techniques.

C. Retrieval

There is a need for search applications to be optimised and reference [21] talks about data retrieval from mind maps which can help enhance search applications. This paper addresses two basic needs of search applications- determining search results and its retrieval; summarizing the contents of result to display on the result page. Various software tools exist for creation of mind maps [22] [23] [24]. In keyword search, a document is considered more relevant if its frequency is more in the document [25]. This is used for ranking of the search results. Common algorithms which use term frequency are TF-IDF [26] and BM25 [27].

The paper by Yizun Wang, Kai Chen, Yi Zhou, Qi Zheng, Haibing Guan [9], discusses about image based retrieval techniques. In such search application, user takes photo and uploads photo to server.

Retrieval is done at the server side and user gets result from the server. The paper proposes an automated offline stable point filtering method for visual search applications. Various transforms to simulate effects caused while taking a photo is discussed and attended to. They are processed in the offline method to reduce the size of the retrieval application. Almost 13% memory or disk space is tried to be saved by loading SIFT features into memory while the application still maintains a high query accuracy.

Retrieval technologies are based on content based image retrieval which is different from old fashioned text annotated image retrieval applications. Text annotated image retrieval is the same as text retrieval and cannot meet the trend because the number of images grows at a very

high rate and manual annotating work takes too much time. Image features extracted automatically would create more realistic annotations. Due to the complexity of digital image, a lot of retrieval methods have been promoted.

The research break-through on distinctive local features such as SIFT [10] and SURF [11] opened up a new direction for image retrieval applications. They provide high quality matches even under severe discrepancies. A CBIR application based on SIFT combined with high dimensional search methods such as locality sensitive hashing (LSH) [12], MP-LSH [13] or other improvements [14] is created. A retrieval example is given but the application requires a lot of memory space and its query speed is sub-linear to the index size. This paper presents an improving method at the offline stage for this application. The index size using stable point filtering method both at the pre-process stage and post-process stage is reduced. Post process utilizes machine learning method to classify retrieval images to decide whether they will be used in stable point filtering. 13.83% space is saved while this application still maintains high retrieval accuracy. This method not only reduces the index size, but also improves the retrieval performance. Many integrated image retrieval systems [15] [16] [17] have been presented before. Their offline processing usually involves the index construction.

Intel suggested a copy-right detection system in 2004. This system uses 6261 gallery pictures as the base database in one of the datasets and demonstrates very high precision and recall. But it expands the 150 query images by transforms. 40 kinds of transforms are made for each query image selected out of the base database. They are all added to the base database to improve accuracy. This system has good accuracy for just 150 images which are similar. This causes increase in cost and space requirement. Although it increases the accuracy it increases redundancy and is not very pragmatic when number of images is huge and memory is less.

CORTINA [16] computes five types of feature descriptors for each image in database. CORTINA uses SIFT feature to do scene classification [18] via pLSA and collect manual annotations from web site tool at the offline stage and queries KD-Tree based index using 12-dimension CFMT feature. This system considers a lot about offline process. But its online query is not sufficient since 12-dimension feature contains only a little information. So, the accuracy will be doubtful if the image is complicated.

Video Google [17] is an object tracking system for videos. It uses clustering method to cluster nearby SIFT descriptors. Each cluster is considered a vocabulary and Video Google uses text approach to do image frame search. When the image database consists of frames from a movie, there will be

consecutive frames that have similar objects. So, the distribution of image features is appropriate for clustering. When one image in the image database has no duplicates or near duplicates, clustering method will generate many poor-quality clusters and both the performance and accuracy will lower down.

A new stable interest point has been promoted in [19] to reduce feature points in database image. It computes stability for each point in the image. A threshold value is set to filter out points whose stability value is below the threshold value. It utilizes probabilistic pose prediction model to detect whether a point is in the pose of an object. Only points agreeing with one pose will be used in stability analysis. Its idea is quite novel, but its offline process is not fully automated and pose

prediction method could only determine one transform just like RANSAC [20] for all the key points and is not very accurate. Big data analytics could be used to improve this.

D. Dealing With Uncertain Data

High volumes of data like log data, business transactions, charts, images from various different fields can be easily collected from different sources. Hence, there is a need for new forms of processing this data, which is the birth of data science, which aims at preparing this data, cleaning it and using it to draw informed decisions and insights. By applying Big data analytics and mining [32], [33], data scientists can extract implicit, previously unknown, and potentially useful information from Big data. Existing Big data mining algorithms [30], [31], [34], [35] mostly focus on association analysis enabled by mining interesting patterns from precise databases. However, there are situations in which data are uncertain.

In real life applications, like the one we are focusing on, a large amount of uncertain data can be produced which might have valuable information or patterns hidden in it. The uncertainty is partially due to inherent measurement inaccuracies, sampling and duration errors, network latencies, or intentional blurring of data to preserve anonymity. In reference [29], a data science solution is presented that uses MapReduce to mine uncertain Big data for frequent patterns according to user constraints. An advantage of using MapReduce is that users can focus on just the map and reduce functions and do not have to worry about the implementation details.

In the recent past, various algorithms have been proposed to use the MapReduce model-which mines the search space with distributed or parallel computing for different Big data mining and analytics tasks [36]. Simple examples of these tasks include clustering [37], outlier detection [38], and structure mining [39]. The most important Big data mining and analytics task is frequent pattern mining, which discovers interesting knowledge in the forms of frequently occurring sets of merchandise items or events. Since the advent of frequent pattern mining [40], various studies have been conducted to mine frequent patterns from precise data traditional. However, data in many real-life applications are riddled with uncertainty [41], [42], [43]. It is partially due to various inherent measurement inaccuracies, duration and sampling errors, network latencies, and intentional blurring of data to preserve anonymity. Hence users are usually uncertain about the presence or absence of items. To handle these uncertain data, various pattern mining algorithms have been proposed. In many real-life applications, users may have some particular phenomena in mind on which to focus the mining However, the above-mentioned algorithms mine patterns without user focus. Therefore, users often need to wait for a long period of time for numerous patterns, out of which only a tiny fraction may be interesting to the users. Hence, constrained pattern mining [44], which generally aims to find those frequent patterns that satisfy the user-specified constraints, is needed.

E. Recommendation

Keyword Based Recommendation System [45], proposes a keyword based recommendation system (KBRS), where the user's preferences are indicated by keywords of their searches and messages. They use a user based filtering algorithm to help provide them with recommendations using map reduce programming. The current recommendation generating algorithms have a few shortcomings and need a few changes to make recommendations more specific to users and significant [46]. The advancements discuss efficient methods to represent users' choices in a more efficient and easy way. We can also use textual information to improve the suggestions provided. Here they make use of the past references and

reviews of users to provide them with recommendations. They make use of map reduce programming but using this in our project would cause compatibility issues, and using android studio with R would be an easier and more efficient method. We will be providing recommendations as to what animals are up for adoption in our app based on users' previous searches. There has been a lot of work that has been conducted on this topic.

In [47], Lu et al. suggested a method for adding social contextual information into recommendation systems by incorporating regularization constraints. The experimental results signify that this advances the precision and accuracy of prediction related to review quality. In [48], a technique was proposed by Mei et al. to combine topic modelling and analysis of social media. A lot of mining issues could be solved using this method. An online news recommendation system for Facebook was put forward by ChengXiang Zhai in [49]. Recommendation was based on the keywords which were created manually for a community. The system improved recommendation decisions based on feedback information. But, it is not personalized for individual Facebook user and has a mechanism only for giving recommendations for news articles. Although these methods do seem to be helpful we will not have access to social media feeds and we will be using the activities of our users to predict what the users might want. We will also be providing recommendations to the users based on searches rather than their text messages. Using big data analytics in our application will help reduce the time people spend searching for what they want and will be provided with options specific to their needs.

F. Framework

Java Framework for Search Applications by Jun-jang Jeng, Lev Kozakov and Sophia Lumelsky [50], supports integration of distributed search centric service components, and enables rapid development and comparative evaluation of search applications. The paper is focused on describing the architectural characteristics of java search service framework (JSSF), including concepts, internal mechanisms and structure. The most important parts of JSSF are domain oriented approach, model decomposition, task centric composition and event based integration. The approach used here allows a system to be broken down to units known as service states. A service state is associated with a family of components which is developed independently. The independence allows components to evolve without changes to other components as long as their functionality is not affected. Service states are common to the concept of workflow systems [51].

Although JSSF has a lot of merits these days there are simpler ways of optimising and developing search applications, one of which is using the R studio which is a simple platform that can be used to analyse and search for data using pattern matching and text analysis. It provides 500+ packages that eases a programmer's tasks and thus reduces the effort and time required to do the same.

Even though there has been a lot of research conducted on this topic we do believe that we can optimise the searches a little further in case of mobile applications. Also, currently there are no mobile applications that provide the features we wish to provide in our application. Google provides the list of veterinarians and hospitals in and around a particular area but it is mainly focused on dogs. We want our application to be of aid to users who own pets other than dogs.

SECTION 5.
DESIGN:

Since we will be improving searches on a mobile application, and providing users with suggestions regarding their searches, we will be making use of android studio to create our application. We have chosen android studio because of the following benefits:

1. Faster deployment of fresh builds: Bringing incremental changes to an existing app code or resource is now easier and faster because of Instant Run. Code changes can be witnessed in the emulator or physical device on real-time without restarting the app or building a new APK (Android Application Package file) every time.
2. More accurate Programming: Featuring an intelligent Code Editor equipped with IntelliJ IDEA interface, Android Studio makes code writing and analysis faster, easier and more accurate. Most of the challenging areas have become a cakewalk now.
3. Faster Programming and Testing: The newly introduced emulator is 3x faster in CPU, RAM, & I/O in comparison to its predecessor. The virtual testing environment is faster than a real device and has a user-friendly UI. Sensor controls are effective to read every move of the developers. Developers can drag and drop APKs for quick installation, resize and rescale the window, use multi-touch actions (pinch & zoom, pan, rotate, tilt) and much more.
4. Inclusive app development: Making multiple builds is in the past. We can build for one and test on multiple devices using Cloud Test Lab Integration. Developers can check the compatibility and performance of an app on a wide range of physical Android devices from within Android Studio.
5. Better App Indexing: Promoting is an important component of the app marketing, and Android Studio 2.0 takes it to a new high. The App Indexing feature available in the IDE helps in creating and adding indexable URL links to the app.

All the data regarding the transactions is going to be stored on the cloud because of its various benefits mentioned earlier. The data is going to be stored on a cloud database and retrieved as and when required. Analysis of the data is going to be done by extracting or copying the data into R studio and performing text and predictive analysis. Various conditions that have to be satisfied to ensure a particular transaction is going to be analysed and portrayed. The steps for the process is given below:

Text analysis, descriptive analysis and predictive analysis is what is going to help provide users suggestions and reducing their search time. The process for each is given below:

Text analysis: Text analysis is the process of collecting, representing and processing data to recognise patterns and relations in the data. It has 3 stages: Parsing - where we convert the unstructured data to structured data to enable easy analysis of data; Searching and retrieval – where we identify the key items or words we need to find in order to analyse the data; Text mining – where we use techniques such as data mining to try and determine the patterns and relations among the data.

Descriptive analysis: Here we use text analysis and plots to describe the relations between the data and tell how each attribute or variable affects another.

Predictive analysis: Here techniques and plots such as linear regression and decision trees can be used to determine the future searches or similar transactions that a user wants. We make use of data that is available until the present, to determine decisions or searches in the future.

By using the above analysis techniques, we can improve the search efficiency since we will be filtering out a lot of data so that each user can search for exactly what he wants and is provided with useful suggestions which would help optimise his searches.

SECTION 6.
SEARCH OPTIMISATION:

Our main aim is to integrate mobile application development along with data analytics to help users find the pets they want or the veterinarians they need. We want to monitor their activity and provide them suggestions and notifications whenever we feel that something that interests them comes up. We want to do this by using R studio along with Android studio to ensure search optimisation. Based on the users' searches we would like to predict their future searches and accordingly suggest important notifications in case something of interest arises.

SECTION 7.
APPLICATIONS:

Search optimisation techniques can improve the efficiency of searches in the following fields and can be applied in any such fields where the efficiency of searching for results plays a crucial role:

- Use of BDA to retrieve information about the vets and other pet related products in and around a user's area.
- To analyse and process data about the amount of time students put into studying and other activities and accordingly draw conclusions from their performances.
- To determine the job opportunities available based on your profile and the jobs that have been advertised online.
- Result verification, where the user can check if an algorithm is performing the required way without errors.

SECTION 8.
CONCLUSION:

The paper proposes a conceptual framework that improves search efficiency by combining multiple approaches. It proposes a model to optimise search efficiency by combining mobile application development and data analytics. We have identified the various challenges that we will have to overcome to ensure our goal is met. The various fields in which efficient searches are required are prodigious and there is no limit to its uses.

SECTION 9.
REFERENCES:

[1] Shreya Upadhyay, Vishal Pant, and Shivansh Bhasin, "Articulating the construction of a web scraper for massive data extraction", IEEE 2017
[2] Ram Sharan Chaulagain; Santosh Pandey; Sadhu Ram Basnet; Subarna Shakya, "Cloud based Web Scraping for big data applications", 2017 IEEE International Conference on Smart Cloud (SmartCloud)
[3] Najma Sultana, Smita Paira, Sourabh Chandra, Sk Safikul Alam, "A brief study and analysis of different Searching Algorithms", IEEE 2017
[4] R S Boyer and J Srother Moore, " A fast string searching algorithm",Communication of the association for computing machinery Inc.20(10),1977, vol-20,pp. 762-772.
[5] Phisan Kaewprapha,Thaewa TansarnandNattakan Puttarak,"Network localization using tree search algorithm"IEEE 2016.
[6] www.cs.auckland.ac.nz/~jmor159/PLDS210/niemann/s_man.pdf
[7] en.wikipedia.org/wiki/Best-first_search
[8] Thae Thae Yu Khine,Koji Araki, Daisuke Mitsunaga and Hua-An Zhao" A nearest neighbor search algorithm for LR-LD on high SNR" IEEE 2016.
[9]An Improved Offline Stable Point Filtering Method for Mobile Search Application Yizun Wang1 Kai Chen2 Yi Zhou3 Qi Zheng4 Haibing Guan5, IEEE
[10] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints" International Journal of Computer Vision, 2004. pp. 91-110
[11] Herbert Bay, Tinne Tuytelaars, and Luc Van Goo, "SURF: Speeded Up Robust Features" Computer Vision – ECCV 2006, pp. 404-417
[12] Alexandr Andoni, Piotr Indyk. "Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions" Foundations of Computer Science, 2006. FOCS '05. 47th Annual IEEE Symposium on Oct. 2006 pp. 459 - 468
[13] Qin Lv, William Josephson, Zhe Wang, Moses Charikar, Kai Li "MultiProbe LSH: Efficient Indexing for High Dimensional Similarity Search" Proceedings of the 33rd international conference on Very large data bases, Vienna, Austria 2007 pp. 950-961
[14] Mayank Bawa, Tyson Condie, Prasanna Ganesan, "LSH Forest: SelfTuning Indexes for Similarity Search", International World Wide Web Conference, 2005, pp. 651-660
[15] Yan Ke, Rahul Sukthankar, Larry Huston. "Efficient Near-duplicate Detection and Sub-Image Retrieval" Proceedings of ACM International Conference on Multimedia (MM), 2004, pp. 869 - 876
[16] Elisa Drelie Gelasca, Pratim Ghosh, Emily Moxley, Joriz De Guzman, JieJun Xu, Zhiqiang Bi, Steffen Gauglitz, Amir M. Rahimi, B. S. Manjunath, "CORTINA: Searching a 10 Million + Images Database" Proceedings of VLDB, 2007, pp. 508-511

[17] Josef Sivic, Andrew Zisserman. "Video Google: A Text Retrieval Approach to Object Matching in Videos" Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) Vol.2 pp. 1470-1477
[18] Aditya Vailaya, Mário A. T. Figueiredo, Anil K. Jain, Hong-Jiang Zhang, "Image Classification for Content-Based Indexing" IEEE Transactions on Image Processing 2001, pp. 117 – 130
[19] Matthew Johnson, Roberto Cipolla, "Stable Interest Points for Improved Image Retrieval and Matching", technical report, http://citeseer.ist.psu.edu/760077.html
[20] Martin A. Fischler, Robert C. Bolles "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", Communications of the ACM, June 1981, Vol. 6, pp. 381 – 395
[21] Joran Beel, "Retrieving data from mind maps to enhance search applications", IEEE 2016
[22] Mind-Mapping.org. Software for mindmapping and information organisation. Website, July 2009.
[23] MindJet. MindJet: About MindJet. Website, July 2009. URLhttp://www.mindjet.com/about/.
[24] SourceForge. SourceForge.net: Project Statistics for FreeMind. Website, 2008.
Available:http://sourceforge.net/project/stats/detail.php?group_id=7118&ugn=freemind&type=-prdownload&mode=year&year=2008
[25] S.E. Robertson and K.S. Jones. Relevance weighting of search terms. Journal of the American Society for Information Science and Technology, 27 (3): 129–146, 1976.
[26] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval.Information processing & management, 24 (5): 513–523, 1988.
[27] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference (TREC'94), 1994.
[28] G. Uddin and F. Khomh, "Mining aspects in API reviews," Polytechnique Montr´eal, Tech. Rep., May 2017.
[29] Carson Kai-Sang Leung, and Fan Jiang, "A Data Science Solution for Mining Interesting Patterns from Uncertain Big Data", 2014 IEEE Fourth International Conference on Big Data and Cloud Computing
[30] A. Cuzzocrea, C. K.-S. Leung, and R. K. MacKinnon, "Mining constrained frequent itemsets from distributed uncertain data," Future Generation Computer Systems, 37, pp. 117–126, July 2014.
[31] V. Kalavri, V. Brundza, and V. Vlassov, "Block sampling: efficient accurate online aggregation in MapReduce," in Proc. IEEE CloudCom 2013, Volume 1, pp. 250–257.
[32] A. Kumar, F. Niu, and C. R´e, "Hazy: making it easier to build and maintain Big-data analytics," CACM, 56(3), pp. 40–49, Mar. 2013.
[33] C. K.-S. Leung and Y. Hayduk, "Mining frequent patterns from uncertain data with MapReduce for Big data analytics," in Proc. DASFAA 2013, Part I, pp. 440–455.
[34] C. K.-S. Leung, R. K. MacKinnon, and F. Jiang, "Distributed uncertain data mining for frequent patterns satisfying anti-monotonic constraints," in Proc. IEEE AINA Workshops 2014, pp. 1–6.
[35] M.J. Zaki, "Parallel and distributed association mining: a survey," IEEE Concurrency, 7(4), pp. 14–25, Oct.–Dec. 1999.
[36] T. Condie, P. Mineiro, N. Polyzotis, & M. Weimer, "Machine learning for Big data," in ACM SIGMOD 2013, pp. 939–942. [5] R.L.F. Cordeiro, C. Traina Jr., A.J.M. Traina, J. L´opez, U. Kang, & C. Faloutsos, "Clustering very large multi-dimensional datasets with MapReduce," in ACM KDD 2011, pp. 690–698.
[38] A. Koufakou, J. Secretan, J. Reeder, K. Cardona, & M. Georgiopoulos, "Fast parallel outlier detection for categorical datasets using MapReduce," in IEEE IJCNN 2008, pp. 3298–3304.
[39] S. Yang, B. Wang, H. Zhao, & B. Wu, "Efficient dense structure mining using MapReduce," in IEEE ICDM Workshops 2009, pp. 332–337.
[40] R. Agrawal & R. Srikant, "Fast algorithms for mining association rules," in VLDB 1994, pp. 487–499.
[41] C.K.-S. Leung & F. Jiang, "Frequent itemset mining of uncertain data streams using the damped window model," in ACM SAC 2011, pp. 950– 955.
[42] C.K.-S. Leung & F. Jiang, "Frequent pattern mining from time-fading streams of uncertain data," in DaWaK 2011 (LNCS 6862), pp. 252–264.
[43] Y. Tong, L. Chen, Y. Cheng, & P.S. Yu, "Mining frequent itemsets over uncertain databases," PVLDB, 5(11): 1650–1661, July 2012.
[44] R.T., Ng, L.V.S. Lakshmanan, J. Han, & A. Pang, "Exploratory mining and pruning optimizations of constrained associations rules," in ACM SIGMOD 1998, pp. 13–24.
[45] Keyword Based Recommendation System by Sandra Elizabeth Salim, R. Jebakumar, IEEE
[46] Carson Kai-Sang Leung, Richard Kyle mackinnon"Reducing the Search Space for Big Data Mining for Interesting Patterns from Uncertain Data", IEEE International Congress on Big Data, 2014.
[47] Chun-Xiao Jiaa,Run-Ran Liua, Tao ZhouP "Personal Recommendation via Modified Collaborative Filtering" 89.75.Hc, 87.23.Ge, 05.70.Ln, 2013
[48] Q. Mei, D. Cai, D. Zhang, and C. Zhai."Topic modeling with network regularization".In Proc. Of WWW '08, pages 101–108, China, 2008.
[49] Manish Agrawal, Maryam Karimzadehgan, chengxiangZhai, "An Online News Recommender System for SocialNetworks", SIGIR-SSM, July 2009.
[50] Java Framework for Search Applications by Jun-jang Jeng, Lev Kozakov and Sophia Lumelsky, IEEE
[51] F. Leymann and D. Roller. Workflow-based Applications. IBM Systems Journal, 36(1):102–123, 1997.
[52] W. Frakes, R. Prieto-Díaz, and E. Fox, "DARE: Domain Analysis and Reuse Environment," Annals Software Eng., vol. 5, Sept. 1998, pp. 125–141.
[53] D. Weiss and R. Lai, Software Product Line Engineering, Addison-Wesley, 1999.
[54] K. Kang et al., Feature-Oriented Domain Analysis (FODA) Feasibility Study, tech. report CMU/SEI-90-TR21, Software Eng. Inst., 1990.
[55] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, 1999.

[10] J.C. Cleaveland, "Building Application Generators," IEEE Software, vol. 5, no. 4, July/Aug. 1988, pp. 25–33.
[56] Y. Smaragdakis and D. Batory, "Application Generators," Dept. Computer Sciences, Univ. Texas at Austin, 1999; www.cc.gatech.edu/~yannis/generators.pdf.
[57] Omar Alonso, "Generating Text Search Applications for Databases", Oracle,2016