Trie-Based LAN File Search Engine

Kartikeyan TR
Department of Computer Technology
Anna University
Chennai
kartikeyantr@gmail.com

Suriyaa V
Department of Computer Technology
Anna University
Chennai
suriyaa2002@gmail.com

Guru Raman C
Department of Computer Technology
Anna University
Chennai
gururaman.c@gmail.com

***Abstract*—Searching and Accessing files in a Computer device is one of the most important and basic task performed by the computer today. Each Operating System has a dedicated search bar for this, But however, In a network with large number of systems, to access any random file from multiple systems is a very challenging job, in other words often it is not possible for the user to remember exactly where he has stored a particular file. This project provides a user friendly application window in which the user enters the name of the file and gets the exact path of the file and the IP address of the Computer Device in the network as output. We can access the file by clicking on the path generated. Accessing Files from multiple systems in a network is quite difficult and a Slow process and there are not many Applications which serve this purpose. We aim to overcome this problem by providing an efficient Trie Data Structure based Searching Algorithm to search and access files in a network from any node in the Local Area Network.**

***Index Terms*—LAN , IP , File Search Engine , Trie**

I. Introduction

Implementing a File Searcher in a Network has always been a difficult process. Search engines work for the own computer rather than many devices in a LAN. In Windows, Searching for files can be done quickly by clicking the Start button at the bottom left of the screen. Then, simply by typing the full or partial name of the file, program or folder. The search begins as soon as we start typing, and the results will appear above the search field. If the file or program being searched for may not appear right away sometimes and latency is produced, as the search can take a while. Grouping of search Results can also be done to obtain similar files.

Our project File Search Engine for networking computers is a software which enables the user to have instant information about any file in that computer and other computers in the network.The software lets us to know the locations of not only that particular file but also the files with similar names on the network so that the file can be searched even if its exact name is not known. The plan also enables one to find out their extensions. One need to just mention the name or any substring of the file and the software would tell the path along with their extensions. Our plan also includes the suggestion facility in which it keeps on suggesting the possible files as the user enters any character. In short this plan is a handy, user-friendly software with the help of which even a person who just knows to operate a computer can find the location of any file.

We pack the query entered by the user into a standard format and searching is done using an Efficient Trie based Algorithm that makes searching easier and more efficient. We are planning on setting up the search engine in the LAN-connected computer environment so that files can be searched on any computer from a single computer on the network. In the future, we also plan on combining AI and voice search into the system to make the search more efficient and easier.

Our software is basically a standalone system application software. So, the market of this software is obviously huge covering all the computer users. Any computer user using this software can have instant access to any information about the files stored in hard-disk.

User needs an internet connection to access this software, the application being lightweight can work smoothly in desktop with very low configuration. We plan to extend the application and scope of the software to LANs so that the users of the systems connected to LANs can know which file is stored in which system.

II. Related Works

A. Research and Implementation of File Search Engine

Under the situation of more and more FTP site resources in the education network, the fast file search capability is important. We are optimizing the file search using an inverted index technology to achieve a prototype system of high-performance FTP search engine as proposed in [1]. The file engine used an brute force algorithm and didn't't do it for a network of computers. These search engines traversal web in a certain strategy using software robot, which is also known as a spider or a crawler, download web documents to the local document libraries for analysis, build up index by the indexer, retrieve the index database in accordance with query requests accepted from user interface, and finally output the query results to users. They own a huge full-text indexing database, mass information and high frequency of updates, are suitable for retrieving artificial information request. But their results are too many their information accuracy is very low, we must filter these results to find information we need.

B. Usage of a binary integrated spell check algorithm for an upgraded search engine optimization

Search engines are the most vital means of accessing information on the Web, where millions of people search

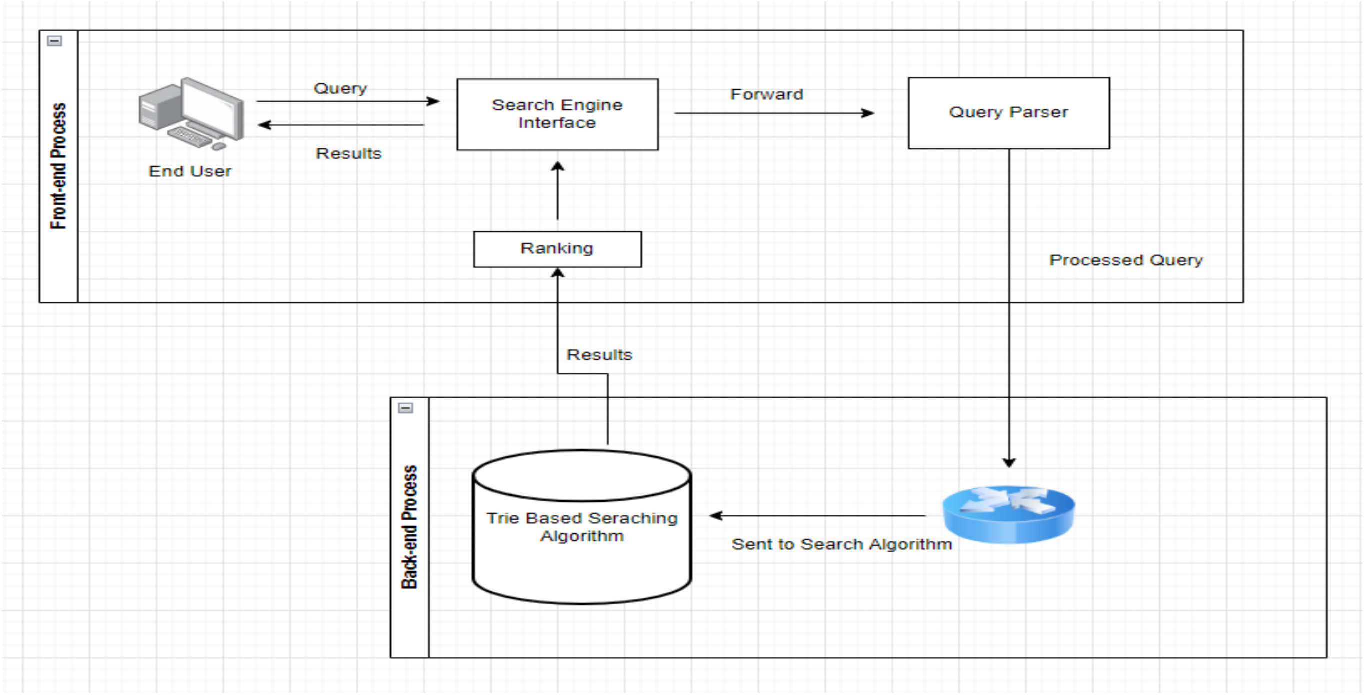


Fig. 1. System Architecture Diagram

to obtain the required information. Statistical data show that approximately 6 billion searches occur every day. However, recent studies also show that the increasing popularity of a number of search tools also increases spell errors or the misspelling of words in a given query. Now, this has become a quite a common problem, with the result that when a search hit occurs place for wrongly spelt words, the results are either wrong or uncertain. It is estimated that such searches of miss pelt words happen 10 to 12 percent of the time and as a result, there are a large number of web pages with miss pelt terms. There much so that our daily effort of retrieving information occurs in an environment (Web Pages) full of broken sentences and queries. A survey on search engines showed that even if a query is misspelled, the authoritative web pages also contain misspelled words, which means that even with the misspelled query, the search retrieves the Web pages with those containing misspelled words. In August 2008, Google was the first search engine to include the Google Suggest feature (later called Auto complete), which helps in formulating queries, reducing errors and keystrokes.

Essentially, a spell check must be able to determine the finest option among all suggestions for misspelled words, instead of providing a list of choices. Web search engines give special attention to formulating queries, upon which the interactive correction mechanism overloads the users even more. This is not an acceptable choice. Therefore, automation in making the right choice of correction would be a better way out.

Offline search tools are not very popular. For example, in any of the Operating Systems used currently, the search for data from the system dump is always more random looking for a particular word that is present or not, retrieving all files or documents or any other data that is present in a meaningless manner, instead of being orderly and semantic. This work [2] presents the evolution of the spell check component of an offline search engine which was developed for retrieving data from archived data. At first, a query is checked for wrong spelling and only then the retrieval of results happens according to the best phase. If the retrieved data is still incorrect, a link is provided to see for possible correct suggestions. The aim of the proposed work is to make offline search engines more accessible to use. To provide satisfactory results with the spell check mechanism, to use a binary converter so that the query correction and evaluation become even more accurate.

C. Research on knowledge search engine based on personalized archives

In the full-text retrieval technology, the traditional keyword information full-text retrieval technology mainly carries out information retrieval for users through the retrieval method matching with the keyword information. It often can only carry out a mechanical information matching for users, which eventually leads to a variety and clutter of retrieval results, Users still need to continue searching in the search results to find their main target search information. Moreover, the retrieval statements entered by users often contain many logical relationships. If these logical relationships are used and added to the additional conditions of retrieval, the accuracy of file retrieval will enter a new level. However, the current file retrieval system often does not support semantic level retrieval, so it is unable to understand the user's retrieval purpose and real intention. This paper [3] improves the accuracy of user search through user preference model and synonym conversion model. In order to increase the accuracy of the search engine.

D. The Anatomy of a Large-Scale Hyper textual Web Search Engine

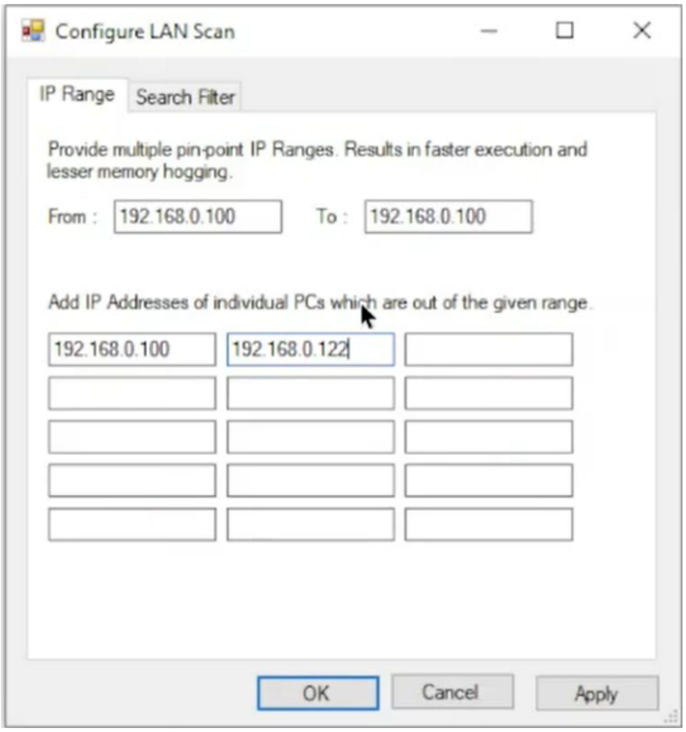
One of the most infamous piece of work was published in [4] In this research, Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext was presented. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. To engineer a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens of millions of queries every day. Despite the importance of large-scale search engines on the web, very little academic research has been done on them. Furthermore, due to rapid advance in technology and web proliferation, creating a web search engine today is very different from three years ago. This paper provides an in-depth description of our large-scale web search engine – the first such detailed public description we know of to date. Apart from the problems of scaling traditional search techniques to data of this magnitude, there are new technical challenges involved with using the additional information present in hypertext to produce better search results. This paper addresses this question of how to build a practical large-scale system which can exploit the additional information present in hypertext. Also we look at the problem of how to effectively deal with uncontrolled hypertext collections where anyone can publish anything they want.

E. Implementation of a Search Engine

With the advent of the web, the information storage and retrieval have taken a huge step forward [5] Search engines plays important role in this area. In this report (implementation of a search engine), we talk about the functionality of a mini-offline search engine. We study the various components of search engine is which involves "crawlers" (a spider program to search through documents), "porter and stemmer" (program that remove stop words and brings the query in its basic form) and "indexer" (one which indexes the documents to cut short the duration of searching). Next part is the implementation of these various components .The search engine while searching through the web gives us so many relevant and irrelevant. Which relevant information should come as a best desired result, depends on the kind of algorithms that all of the search engine have got the propriety right over. I have also tried to develop a similar algorithms named as page rank, which has been implemented on a small web graph and extract the information through the in-links and out-links of any web page.

III. Current System

- In the existing system of file search engines, it is hard to look at information and search for data on multiple systems at a time.
- A query is passed as is then searched. This way of searching can lead to many errors and is also an inefficient way of searching.
- The latency time to search for the files is very high especially if searched in a network.



Provide multiple pin-point IP Ranges. Results in faster execution and lesser memory hogging.

From : 192.168.0.100 To : 192.168.0.100

Add IP Addresses of individual PCs which are out of the given range.

192.168.0.100 192.168.0.122

Fig. 2. IP Configuration Interface

IV. Proposed Trie Search

- Packing the query entered by the user into a standard format and searching is done using an Efficient Trie based Algorithm that makes searching easier and more efficient.
- Producing the result as a path link rather than just a static text name to enable accessing file directly from the Search Application itself.
- Plan to set up the search engine in the LAN-connected computer environment so that files can be searched on any computer from a single computer on the network.
- Preference based searching of files in the computer with the help of search and extension enabled filters.
- Pattern matching of files of similar types and name is done and provided to the user

Future Improvements also include the plan of combining Artificial Intelligence (AI) and voice based search into the system to make the search more efficient and easier without any hassle and accessible to everyone.

V. Local Area Network (LAN)

A local area network (LAN) is a group of computers and peripheral devices that share a common communications line or wireless link to a server within a distinct geographic

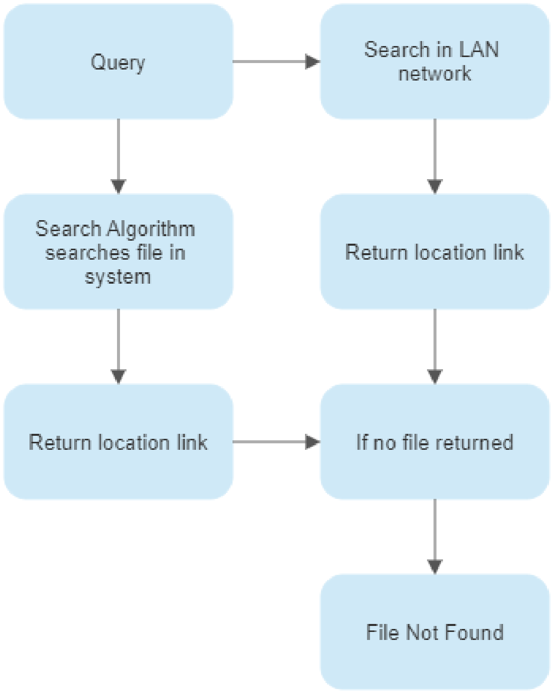


Fig. 3. Search Interface

area [6]. A local area network may serve as few as two or three users in a home office or thousands of users in a corporation's central office. Homeowners and information technology (IT) administrators set up LANs so that network nodes can communicate and share resources such as printers or network storage.

LAN networking requires Ethernet cables and Layer 2 switches along with devices that can connect and communicate using Ethernet. Larger LANs often include Layer 3 switches or routers to streamline traffic flows.

A LAN enables users to connect to internal servers, websites and other LANs that belong to the same wide area network (WAN) as introduced in [8]. Ethernet and Wi-Fi are the two primary ways to enable LAN connections. Ethernet is an Institute of Electrical and Electronics Engineers (IEEE) specification that enables computers to communicate with each other. Wi-Fi uses radio waves in the 2.4 gigahertz and 5 GHz spectrum to connect computers to the LAN.

Legacy LAN technologies, including token ring, Fiber Distributed Data Interface (FDDI) and Attached Resource Computer Network (ARCNET) have lost favor as Ethernet and Wi-Fi speeds increased and connectivity costs decreased.

Understanding local area networking There are two primary LAN types: wired LANs and wireless LANs (WLANs). A wired LAN uses switches and Ethernet cabling to connect endpoints, servers and internet of things (IoT) devices to the corporate network [7]. For small businesses with only a handful of devices, a wired LAN can consist of a single un-managed LAN switch with enough Ethernet ports to interconnect all devices. But larger LANs that connect thousands of devices require additional hardware, software and configuration steps to ensure the network is performing optimally. This is where the concept of virtual LANs (VLANs) comes into play.

Because an Ethernet LAN is a shared medium, if an organization has too many devices connected to a single LAN, the amount of broadcast traffic – which is heard by all devices on the LAN – can create congestion and bottlenecks. To alleviate the amount of broadcast traffic being sent and received on a LAN, the network can be broken into multiple VLANs. This condenses the broadcast traffic so it's only heard by other devices within that virtual LAN – not the entire network. This eliminates much of the broadcast overhead that can lead to performance problems.

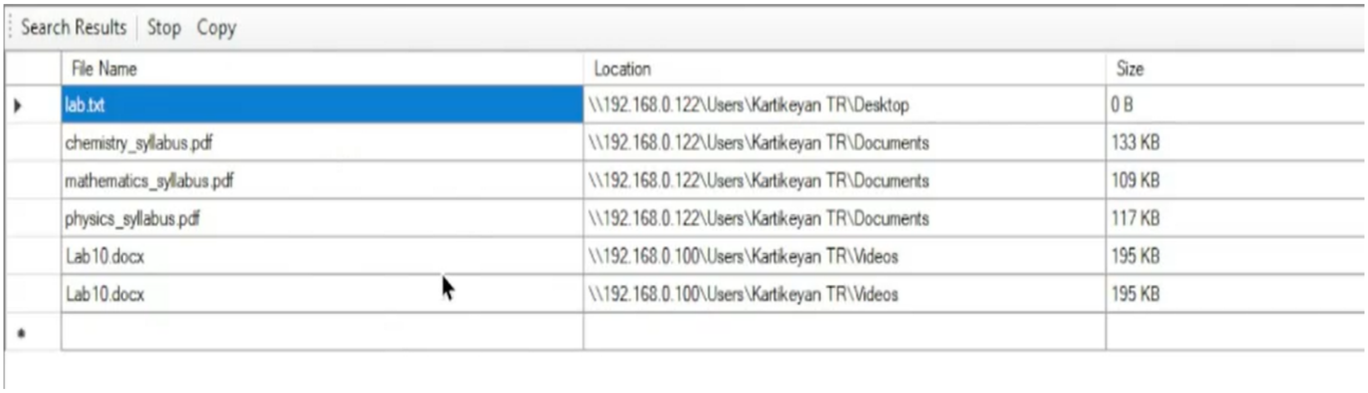
Although virtual LANs can help reduce broadcast congestion issues, they create another problem. When devices on different VLANs need to talk to each other, a Layer 3 switch is required to transmit and receive traffic between the two LANs. This is known as inter-VLAN routing. Additionally, because large enterprise networks almost always are broken up into hundreds of VLANs, they require routers to be deployed throughout parts of the overall network. Today, vendors integrate Layer 3 routing capabilities into network switches to create a Layer 3 switch. Thus, a Layer 3 switch can perform both switching and inter-VLAN routing functions on a single appliance.

Wireless LANs use the IEEE 802.11 specification to transport data between end devices and the network using wireless spectrum. In many situations, a wireless LAN is preferable to a wired LAN connection because of its flexibility and cost savings, as it isn't necessary to run cabling throughout a building. Companies assessing WLANs as a primary means of connectivity often have users who rely exclusively on smartphones, tablets and other mobile devices.

Setting up a basic local area network Operating systems (OSes), such as Microsoft Windows, Linux, Apple OS X, Android and iOS, have Internet Protocol Version 4 (IPv4) and IPv6 networking capabilities incorporated into them. Additionally, personal computer (PC), tablet and smartphone hardware all come with an Ethernet port, Wi-Fi chip or both. This means that, as long as the network administrator has a relatively up-to-date laptop or desktop PC, it's fairly straightforward to network machines together onto a wired or wireless LAN.

Setup of a simple wired LAN requires an administrator to connect the end device to a LAN switch using a twisted-pair Ethernet cable. Once connected, the devices can communicate with each other on the same physical LAN or VLAN.

To set up a wireless network, the administrator needs a wireless access point (WAP). The WAP can be configured to broadcast a network service set identifier (SSID) and require devices to authenticate to the network using one of several Wi-Fi authentication techniques. Popular authentication options include Wi-Fi Protected Access 2 pre-shared key and WPA2 Enterprise.



File Name	Location	Size
lab.txt	\\192.168.0.122\Users\Kartikeyan TR\Desktop	0 B
chemistry_syllabus.pdf	\\192.168.0.122\Users\Kartikeyan TR\Documents	133 KB
mathematics_syllabus.pdf	\\192.168.0.122\Users\Kartikeyan TR\Documents	109 KB
physics_syllabus.pdf	\\192.168.0.122\Users\Kartikeyan TR\Documents	117 KB
Lab10.docx	\\192.168.0.100\Users\Kartikeyan TR\Videos	195 KB
Lab10.docx	\\192.168.0.100\Users\Kartikeyan TR\Videos	195 KB

Fig. 4. LAN Search Results

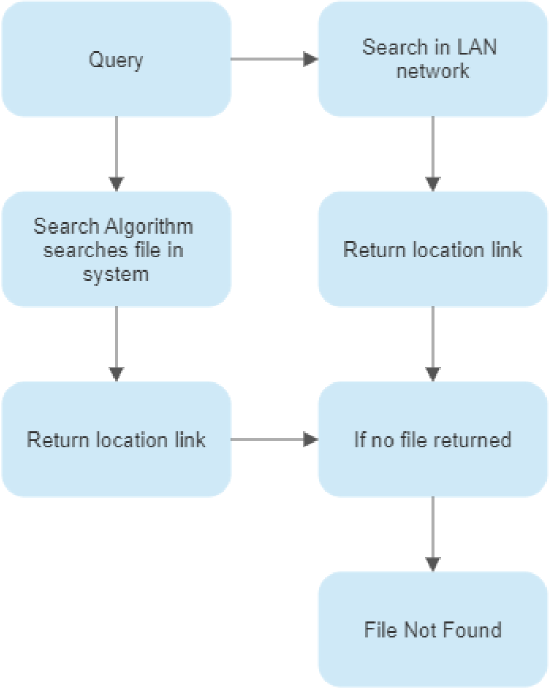


Fig. 5. Flow Diagram

VI. Implementation

The implementaion starts with creating User Interface. The user interface is created using C design forms. We implemented 3 forms. One for Main interface where we search and results are shown. The next form is for settings to configure the IP addresses. The nect from is for the about section. In the backend we used the .NET framework to implement the file fetching in LAN.

The application works by specifying the IP range in which the search engine works by entering a range of IP address. The tool searches for live hosts. It then determines the list of folders available that is shared by the systems and looks for files (according to a keyword that we specify) and displays all the files with similar matching names and displays it on the application window. From here, the file can easily be accessed by just clicking on it.

Determining whether a system with a particular IP address is active. This was achieved using the Ping class defined in the namespace System.Net.NetworkingInformation. The Ping class offers both synchronous and asynchronous methods to detect whether a remote host is reachable. The important part of implementation is TRIE. Every node of Trie consists of multiple branches. Each branch represents a possible character of keys. Mark the last node of every key as the end of the word node. A Trie node field isEndOfWord() is used to distinguish the node as the end of the word node. Inserting a key into Trie is a simple approach. Every character of the input key is inserted as an individual Trie node. Note that the children is an array of pointers (or references) to next-level trie nodes. The key character acts as an index to the array children. If the input key is new or an extension of the existing key, construct non-existing nodes of the key, and mark the end of the word for the last node. If the input key is a prefix of the existing key in Trie, Simply mark the last node of the key as the end of a word.Searching for a key is similar to the insert operation. However, It only compares the characters and moves down. The search can terminate due to the end of a string or lack of key in the trie. In the former case, if

Algorithm 1 File Search

```
for Every query do
    Parse the query
    if Query is Valid then
        Send parsed query to LAN network
        Result = TRIE(Query)
        if Result == true then
            return file location
        end if
        return file not found
    end if
end for
```

the isEndofWord field of the last node is true, then the key exists in the trie. In the second case, the search terminates without examining all the characters of the key, since the key is not present in the trie. Create a root node with the help of TrieNode() constructor.Store a collection of strings that have to be inserted in the trie in a vector of strings say, arr.Inserting all strings in Trie with the help of the insert() function. Search strings with the help of search() function.

VII. Results

The following objectives have been accomplished:

- Enable the user to have instant information about any file in that computer and other computers in the network.
- Obtaining the locations of not only the particular file which we search but also the files with similar names on the network so that the file can be searched even if its exact name is not known.
- Preference based searching of files in the computer with the help of search and extension enabled filters.
- Pattern matching of files of similar types and name is done and provided to the user
- Enable one to find out the file just by mentioning the name or any sub string of the file the software would tell the path along with their extensions.

VIII. Conclusion and Future Works

In this paper, File search has been done successfully for multiple computers in a local area network (LAN). Optimization of the search algorithm using Google recommended method for file search - Trie Data Structure has been proposed. This File Search Engine is found to be highly useful especially University labs and in IT companies. Future work will involve improving the accuracy and efficiency this search algorithm by combining AI and voice search into the system to make the search more efficient and easier. Thus highlighting the future scope of our piece of work.

References

[1] Research and Implementation of File Search Engine; Weihong, D., Guoli, L. IEEE,2019 Electronic ISSN: 2153-0742:

[2] Usage of a binary integrated spell check algorithm for an upgraded search engine optimization;S.Gowri P.J.Sathish Kumar K.Geetha Rani R.Surendran J.Jabez; 2020 IEEE Paper

[3] L. Ye, ”Research on knowledge search engine based on personalized archives,” CIBDA 2022; 3rd International Conference on Computer Information and Big Data Applications, 2022, pp. 1-5.

[4] Brin, Sergey, and Lawrence Page. ”The anatomy of a large-scale hypertextual web search engine.” Computer networks and ISDN systems 30.1-7 (1998): 107-117.

[5] Mirzal, Andri. (2012). Design and Implementation of a Simple Web Search Engine. Int’l J. Multimedia and Ubiquitous Engineering. 7. 53-60.

[6] D. D. Clark, K. T. Pogran and D. P. Reed, ”An introduction to local area networks,” in Proceedings of the IEEE, vol. 66, no. 11, pp. 1497-1517, Nov. 1978, doi: 10.1109/PROC.1978.11152.

[7] A. A. M. Khalaf, M. S. A. Mokadem and K. A. Ahmad, ”Performance of LAN under different ethernet wiring standard and different frame size,” 2015 17th International Conference on Advanced Communication Technology (ICACT), 2015, pp. 638-647, doi: 10.1109/ICACT.2015.7224875.

[8] X. Hu, ”Study on wireless local area network technology,” 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), 2012, pp. 609-612, doi: 10.1109/CECNet.2012.6201526.