International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5S3, July 2019

An Efficient Method to Extract Geographic Information

A. Shiny, Reuma Akhtar, Saurav Singh, K. Sujana, D.V. Neelesh

Abstract: ***A ton of vital geographic information about spots, including points of interest, areas and personal information such as neighborhoods, phone numbers etc. can be found on the Internet. However, such information is not openly available using legitimate means. Furthermore, the given information is temperamental as it is static and not refreshed every now and again enough. In this paper, using the results of an internet list, an effective method to manage and collect datasets of spot names is demonstrated. The strategy proposed is to use the Google web crawler Application Programming Interface in order to recoup site pages related with express territory names and types of spots and after that analyses the resultant website pages to remove addresses and names of places. Using the data gathered from internet, the final result compiled is a dataset of spot names. We survey our philosophy by using accumulated data found using street view of Google Maps by examining signs belonging to businesses found in images. The conclusion exhibited by the results was that the modelled procedure efficiently created spot datasets on par with Google Maps and defeated the results of OSM.***

Index Terms: ***datasets, geographic information, points of interest, spot names.***

I. INTRODUCTION

A large spectrum of geographic applications relies on spot data for structure, for instance, territory-based organizations used by cell phones. Due to these necessities, methods for normally gathering spot name datasets are direly required. For the most part, place datasets can be gained from composed sources, for instance, Google Maps and OpenStreetMaps. They give static information and don't always intertwine the latest changes. In like manner, business sources, for instance, the Google Places API keep up bewildering region data and use a lot of restrictions to limit the amount of information that can be extracted from it. Curiously, vast amounts of spot data unreservedly available on the Web is not only amazingly enormous, it also increases at a rapid pace. Furthermore, the unformatted structure of pages empowers them to change from time to time, and bleeding edge information about spots is much of the time most promptly available on the Web. The test is the way by which to isolate exact and favourable geo data using the Internet to fabricate relevant datasets.

The proposed procedure of this paper uses web crawlers that select appropriate pages for removing and manufacturing spot-name datasets. Our procedure journeys pages using specific zones and type of spot. For instance, "Wall Street, New York, Business Street". Further, we separate spot name and address information from the obtained ordered records. A definitive outcome is a spot name dataset expelled direct from the Internet.

Manuscript published on 30 July 2019.
* Correspondence Author (s)
A.Shiny, Department of CSE, SRM Institute of Science and Technology, Chennai, India.
Reuma Akhtar, Department of CSE, SRM Institute of Science and Technology, Chennai, India.
Saurav Singh, Department of CSE, SRM Institute of Science and Technology, Chennai, India.
K. Sujana, Department of CSE, SRM Institute of Science and Technology, Chennai, India.
D.V. Neelesh, Department of CSE, SRM Institute of Science and Technology, Chennai, India.

II. MODELLING OF PLACE NAME DATASETS

This methodology consists of four parts, specifically, Aggregation of Webpages, Refining and Filtering irrelevant content, Extraction of place information, and Visual Representation. The subtleties of the modules are exhibited in the accompanying sections.

A. Aggregation of Webpages

Associations are used for intended spot types that are to be removed from the web. To assemble relevant business website pages, we at first total a once-over of look catchphrases for scrutinizing a web list API. The web look device request join three areas, to be explicit, Street Names, City Names, and Business Types. To set up our request, we recently set a specific city name, and the Webpage Aggregation module subsequently download data related to streets using OpenStreetMap and concentrates their names using OpenStreetMap "address:streetname" quality (e.g., "Wall Street" and "Manhattan"). For commercial types, we physically set up rundown of pervasive spot types, for instance, Church, Club, Gym, Mall, Bank, etc. Finally, the Aggregation process gives the gathered request to a web list to find commercial pages. Due to being unrivaled in its efficiency, the search engine used to perform the given errand is Google.

B. Refining and Filtering Irrelevant Content

Web crawlers can restore indistinguishable number of results from we need, yet an extensive bit is inconsequential. In the Webpage Aggregation module, the assembled pages join various land postings (which contain generally private areas). Such postings can be refined. The procedure requesting the Google web crawler using a starting late sold road number and a while later find the space names of understood land destinations using the compiled results.

Searching "21 Mirador" in Google for example, returns pages from land locales including Loopnet, Zapmeta, etc. Using these destinations, we become acquainted with the URL instances of land locales, for instance, "www.loopnet.com" and "www.zapmeta.com". Then, the insightful URL guides are used to empty land locales normally.

C. *Extraction of Place Information*

Since site pages have diverse record structures. Isolating location information from these assorted sorts of site pages is trying. For instance, a single area can be expelled using a segment of the accumulated site pages each of which addresses a particular business; The yellow postings represent unique results. Figure 1 exhibits the general procedure for spot data retrieval. In case the first algorithm confirms that the given site page is set to a specific need or purpose, it removes the spot name and an astounding area. Something different, the page is sent to the second algorithm for evacuating various spot names and addresses.

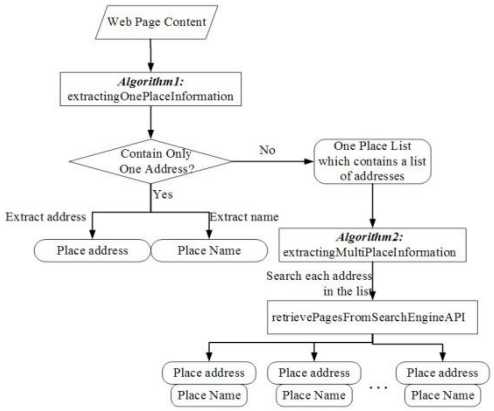

Fig. 1: Geospatial Data Extraction of Places from Webpages

In the Algorithm 1, there can be two scenarios that arise in the area extraction: the first scenario arises when the entire area is stated in a single line; In the second scenario, the area extends over multiple lines.

For the first case, we initially find the address in the site page that starts with a number followed by the street name and city name. In the same manner, the length of the line should be not actually a given edge tolerating that the probability of a line describing an area is commonly less if it is large in size. In our work, the breaking point regard is set to a particular number precisely. In the other scenario, we use a near system to choose if a combination of two lines address one area. For example, if the primary line starts with a number and the next line mentions street name, then two are combined and removed as an area if the length of the joined line is shorter than the cutoff. In the Algorithm 1, if site page has only a solitary area, we expect the site page title is the spot name. If the page contains various areas, the Algorithm 1 passes on an assortment of addresses to the Algorithm 2.

Algorithm 1: extractingOnePlaceInformation

```
Input  :Web Page page, City Name cityName,
        Maximum length of Address threshold
Output : singlePlace<address, name>,
        List multiAddressList
content←FilterHtmlTag(page)
while line1 in content do
  if StartWithNumber(line1) then
    if StrContainCityName(line1, cityName) then
      if theLengthOf(line1)<threshold then
        multiAddressList.append(line1)
    tempValue←line1
  else if StartWithCityName(line1, cityName) then
    line2← catenation(tempValue,line1)
    if StartWithNumber(line2) then
      if StrContainCityName(line2, cityName) then
        if theLengthOf(line2)<threshold then
          multiAddressList.append(line2)
  if addr_list.hasOnlyOneAddress() == true then
      singlePlace.address← multiAddressList.firstElement
      singlePlace.name ← extractingWebPageTitle(page)
      return singlePlace
  else
    return multiAddressList
```

Algorithm 2: extractingMultiPlaceInformation

```
Input  : List addr_list
        //A list of addresses from one page
Output : PlaceList<address, name>
while item in addr_list do
  pList←retrievePagesFromSearchEngineAPI(item)
  //Retrieve the returned pages from search Engine.
  while page in pList
    if page.containOnlyOneAddress==true then
      addr←extractOneAddress(page)
      if addr==item then
        PlaceList.address← addr
        PlaceList.name←extractingWebPageTitle(page)
        break
return PlaceList
```

In the Algorithm 2, our aim is to traverse every location in the location list using the Internet. From the website pages that are returned, if a page has just a single location and the location is equivalent to the location used to scrutinize the web crawler API, the relevant webpage page heading is treated as the spot name. The site page heading along with the isolated area is used to depict the spot.

D. *Visual Representation*

When we remove the spot name and spot address from the previous module, we utilize a geocoding tool to change over the spot address into geographic coordinate points for envisioning the removed spot datasets. Google Fusion Tables is a web service provided by Google for data management. Fusion tables can be used for gathering, visualising and sharing data tables. As represented in fig. 2, we changed two sorts of pin locations. The pin locations with mark 'y' address the spot name with business information. The pin locations with mark 'n' mean those spots accessible to be obtained or rent, which are filtered through by the Filtering Irrelevant Content module.

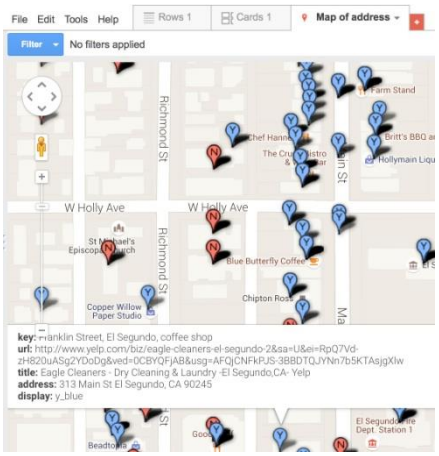

Fig. 2: Coding of Geographic Information and the Visualization of Compiled Datasets

III. EXPERIMENTAL RESULTS

This area depicts our experimentation system and describes the outcomes of our research. We gathered spot-name datasets from the Web for New York City utilizing the technique and matched the outcome with OSM and Google Map. We compared the dataset with respect to Precision, Recall, and F-score.

A. *Experimental Settings*

In our examinations, we initially used Google Search to explore vital commercial webpage for New York City. Table 1 shows the resultant addresses produced by the experiment. Using OSM, street names were obtained. We picked ten squares in New York City to assess the execution of the displayed system.

B. *Evaluation Methods*

Table 1: Place Address Datasets

To assess our methodology, we physically took a gander at

Street Names (in New York) ***(Extracted from OpenStreetMap)***	Houston Street
	Avenue of the Americas
	Canal Street
	Wall Street
	Broadway
	Madison Avenue
	Steinway Street
	Christopher Street
	Barrow Street
	Bond Street
	Love Lane
	Gold Street
City Names	New York
Place Types	College/Café/Stadium/ Gym/ Mall/ Airport/ Amusement park/ ATM/ Resort/ Mosque/ Spa

GoogleStreetView to gather the actual information by perusing signs noticeable in the symbolism. The reality of the situation could prove that Google Street View itself was not cutting-edge, yet this fleeting predisposition should influence each of the three test measures. The Precision, Recall, and the F-Score values were utilized to assess the exhibitions of the displayed methodology.

C. *Experimental Result*

Table 2 demonstrates a case of the extricated spots from the Internet utilizing our methodology, place from Google Map and OSM. In this block, OSM showed just a single spot name. Our separated dataset contains an extra spot than Google Map. The yellow line demonstrates an accurately removed spot utilizing our methodology, which was absent in the Google Maps dataset.

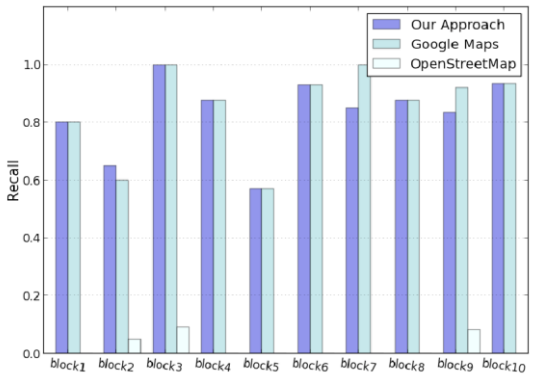

Fig. 3: Recall comparision between our method, OSM, and Google Map

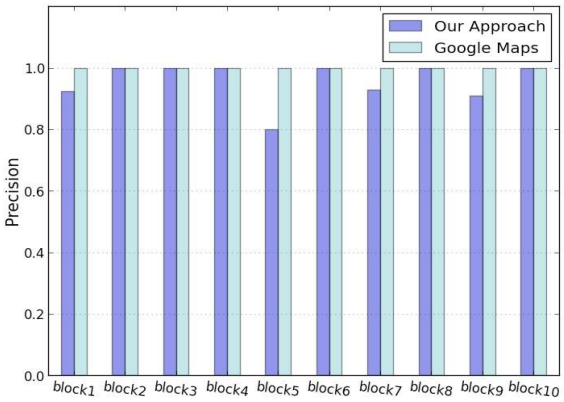

Fig. 4: Precision comparision between our method and Google Map

Table 2: Sample results returned for the block of one city.

Our approach		Google Maps		Google Street View (Ground Truth)		OpenStreetMap	
Place Name	Address	Place Name	Address	Place Name	Address	Place Name	Address
Main StreetRealty - Real Estate Agents	361 Main St El Segundo, CA	Main StreetRealty	361 Main St El Segundo, CA	Main Street Realty	361 Main St El Segundo, CA	---	---
On The Main-El Segundo	353 Main St El Segundo, CA	On the main	353 Main St El Segundo, CA	On the main	353 Main St El Segundo, CA	---	---
Blue Butterfly Coffee Co - Coffee & Tea	351 Main St, El Segundo, CA	Blue Butterfly Coffee	351 Main St, El Segundo, CA	Blue Butterfly Coffee Company	351 Main St, El Segundo, CA	---	---
Chipton Ross in El Segundo | Chipton Ross	343 Main St El Segundo, CA	Chipton Ross	343 Main St El Segundo, CA	Chipton-Ross	343 Main St El Segundo, CA	---	---
The Jewelry Source - Jewelry	337 Main St El Segundo, CA	Jewelry Source	337 Main St El Segundo, CA	The Jewelry Source	337 Main St El Segundo, CA	---	---
The Original Rinaldi's Sandwiches	323 Main St El Segundo, CA	The Original Rinaldi	323 Main St El Segundo, CA	The Original Rinaldis Italian Dcli	323 Main St El Segundo, CA	---	---
Steve's Burgers - Burgers	321 Main St El Segundo, CA	Steve's Burgers Plus	321 Main St El Segundo, CA	Steve's Burgers Breakfast¡¤Lunch¡¤ Dinner	321 Main St El Segundo, CA	---	---
Eagle Cleaners - Dry Cleaning & Laundry	313 Main St El Segundo, CA	Eagle Cleaners	313 Main St El Segundo, CA	Cleaners eagle	313 Main St El Segundo, CA	---	---
World Karate - Martial Arts	309 Main St El Segundo, CA	---	---	World Karate	309 Main St El Segundo, CA	---	---
Bank of America (ATM)	343 Main St El Segundo, CA	Bank of America (ATM)	343 Main St El Segundo, CA	Bank of America ATM	343 Main St El Segundo, CA	Bank of America	343 Main St El Segundo, CA

Table 3: Final results including entire list of city blocks.

Block No.	Our Approach		Google Street View (Ground Truth)	Google Maps		OpenStreetMap	
	Total number of the extracted records	Number of the exact records	Total number in Ground Truth	Total number of the extracted records	Number of the exact records	Total number of the extracted records	Number of the exact records
1	13	12	15	12	12	0	0
2	13	13	20	12	12	1	1
3	11	11	11	11	11	1	1
4	7	7	8	7	7	0	0
5	5	4	7	4	4	0	0
6	11	11	13	11	11	0	0
7	14	13	14	14	14	0	0
8	7	7	8	7	7	1	0
9	11	10	12	11	11	1	1
10	14	14	15	14	14	0	0

Table 3 shows general outcomes for every test square. We contrasted our methodology with Google Map and OSM with respect to the Recall, Precision, and F-Score values on the squares to be tested. Figure 3 demonstrates that the Recall of our methodology was a lot greater than OSM. Furthermore, it was equivalent to or marginally lesser than Google Map for almost all the test squares aside from square 2 (where our strategy outflanked Google Maps).

The average Precision values for Google Map, OpenStreetMap and our methodology was 100%, 75%, and 95.6%.. Figure 4 shows that the Precision of our methodology was as efficient as Google Map. Since OpenStreetMap returned no place in the vast majority of the test squares, we did not demonstrate its accuracy here.

For a place such as Ney York City, business data is hardly available on OpenStreetMap, our outcomes could be utilized to give a beginning point for outsourcing.

Contrasting with our methodology, Figure 5 demonstrates the quantity of the spots absent on OSM in the squares.

Figure 6 demonstrates that our methodology outflanked OSM in terms of F-Score and acquired a comparable F-score as Google Map. The Precision value of our methodology was lesser than Google Map when the quantity of the extricated spots was increased. The false-positives in this methodology were because of the below circumstances. To start with, we utilized the website headings as the spot names, yet a couple of the extricated places utilized locations as their web page headings.

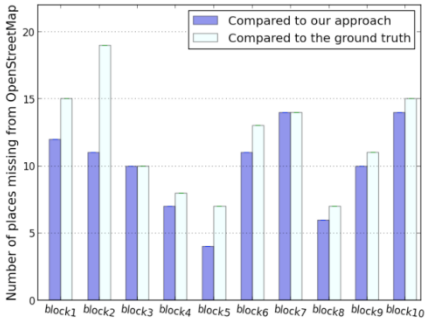

Fig. 5: Total amount of missing data for OSM compared to this paper's approach.

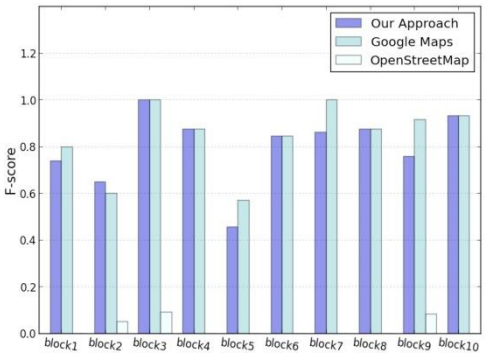

Fig. 6: Comparison of statistical F-Score deviations between our results, OSM, and Google Map.

Around 65% of the inaccurate outcomes were brought about by the non-place-name web page headings. Secondly, we extricated locations from small lines that begin with a number followed by street names. The vast majority of the addresses fulfilled this standard, yet there were a few special cases. For instance, our methodology inaccurately extricated the spot name "Lot-Less Store - Discount Store - NYC, NY" as a location. Around 30% of the inaccurate outcomes were brought about by having inaccurate address. We could solve the first issue by sifting through such irrelevant addresses, that are in the type of a location. To solve the other issue, we could include extra requirements, for example, regardless of whether the address line contains a road name to decide the right address.

On comparison with the ground truth, 80% of the spots that were absent in our methodology were because of the absence of search-query variety. For instance, in our trial, "music scene" was not specified as a spot type in the pursuit question, "Medison Square Garden", "MetLife Stadium", and so forth., were missed in the indexed lists. Whatever remains of the missing places was brought about by missing websites.

IV. RELATED WORK

As the amount of geospatial data on the internet increases rapidly day, abusing freely accessible Internet hotspots for separating place data can help to rapidly and adequately create a huge arrangement of spot data for any city on Earth.

Since geographical data is always changing, the collected datasets need to be updated regularly. This can be done using Web scratching, i.e., going through the raw HTML of the pages for producing corpus that are accurate compared to the most recent spot data on the Web. However, since HTML is presentation based, this extraction proves difficult. Thus, as shown in [1], a wrapper is used which is a tool that extracts data based on the DOM structure of the HTML page.

Various scientists have gathered information for producing geographical datasets and dictionaries using organized sources on the internet. For example, in [2] the methodology defines the top query to find streets of interest and run it through its algorithm using criteria such as spatio-textual relevance and diversity and presents experimental results using three European capital cities as datasets.

Going further, in [3], a method was devised for preparing high-quality geographical data collections from online open data, such as OSM, Geonames, and DBpedia, and deriving a language model from them. Carsten Keßler, et al in [4], Use clustering and filtering algorithms to find place names and then assign them adequate geographic footprints. They then compare the experimental results of their work to results obtained from traditional gazetteers for three geographic locations - Soho,Camino de Santiago and Kil-imanjaro.

However, there are several problems that arise when trying to extract geographic data. One such problem is the fact that users online use vague and imprecise place names to describe locations such as "Midwest US", "UK Midlands" etc. The trouble arises because there are no official set boundaries for such locations and also by the fact that such terms are used more frequently. One proposed method [5] is to find the coordinates of known locations such as cities, towns, etc inside and outside the imprecise location and use those to calculate the boundaries of that area. Similarly, inside a city or a town, neighbourhoods are geographical units used to socially and sometimes economically classify people. Such areas are valuable geographic units upon as policy decisions could be considered based on them. However, geographically, there isn't an official method to classify neighbourhoods. A passive methodology of mining data has been proposed in [7] to extract vague neighbourhood areas. This involves collecting and analysing postal addresses to gather data and derive neighbourhoods from them. And finally doing an analysis to find synonyms for the neighbourhood.

A geographical dictionary or a gazetteer is a collection that contains a list of geographic data consisting of three main components; the name, type, and footprint. It converts information usually presented in vernacular language to a scientific one. Using the vast amount of data present on the internet, one can arguably create a highly detailed gazetteer of geographic information. One such approach to creating a gazetteer is using semi-supervised learning as shown in [9] where a combination of manual matching and supervised learning allowed a substantially larger proportion of queries to be classified than any other technique. Thus, building on this, the model presented in [8] uses semi-supervised learning to generate a set of geographic features and fill each item in the set with its associated name, type and geographic footprint.

V. CONCLUSION AND FUTURE ENHANCEMENTS

We have introduced a methodology for modelling place name datasets from the Web in this paper. Although, the exactness of our methodology may be less precise compared to google maps, our assessment demonstrated that when the quantity of extricated spots was huge, the review of our methodology was dependably similar to or higher than Google Maps.

We have found three fundamental bearings based on our current and future work. The precision of the data extraction module needs to be improved. This should be possible by abusing the Document Object Model used by website pages. Next, extracting catchphrases of spot types using the internet can be done to refine the review. Similarly, taking a preferred standpoint of the cosmology ideas to characterize an adaptable method will build up semantically strong connections between spot types. Such strong connections can improve the review by giving improved look inquiries to internet searcher. In conclusion, we intend to lead an increasingly broad test that covers urban communities of different sorts on the planet.

REFERENCES

1. Raeymaekers S., Bruynooghe M., Van den Bussche J. Learning (k,l)-Contextual Tree Languages for Information Extraction. In: Gama J., Camacho R., Brazdil P.B., Jorge A.M., Torgo L. (eds) Machine Learning: ECML 2005. Lecture Notes in Computer Science, vol 3720. Springer, Berlin, Heidelberg, 2005.
2. Skoutas, Dimitrios, Dimitris Sacharidis, and Kostas Stamatoukos. "Identifying and Describing Streets of Interest." In EDBT, pp. 437-448, 2016.
3. Keßler C, Maué P, Heuer JT, Bartoschek T. Bottom-up gazetteers: Learning from the implicit semantics of geotags. In International Conference on GeoSpatial Sematics (pp. 83-102). Springer, Berlin, Heidelberg, 2009.
4. Al-Olimat HS, Thirunarayan K, Shalin V, Sheth A. Location name extraction from targeted text streams using gazetteer-based statistical language models. arXiv preprint arXiv:1708.03105, Aug 2017
5. Arampatzis A, Van Kreveld M, Reinbacher I, Jones CB, Vaid S, Clough P, Joho H, Sanderson M. Web-based delineation of imprecise regions. Computers, Environment and Urban Systems, 2006.
6. Zhang Y, Chiang YY, Knoblock C, Zhang X, Yang P, Gao M, Ma Q, Hu X. Extracting geographic features from the Internet: A geographic information mining framework. Knowledge-Based Systems, Mar 2019.
7. Goldberg DW, Wilson JP, Knoblock CA. Extracting geographic features from the internet to automatically build detailed regional gazetteers. International Journal of Geographical Information Science, Jan 2009.
8. Brindley P, Goulding J, Wilson ML. Generating vague neighbourhoods through data mining of passive web data. International Journal of Geographical Information Science, Mar 2018.
9. Beitzel SM, Jensen EC, Frieder O, Lewis DD, Chowdhury A, Kolcz A. Improving automatic query classification via semi-supervised learning. In Fifth IEEE International Conference on Data Mining (ICDM'05), Nov 2005.
10. Nguyen HT, Cao TH. Named entity disambiguation: A hybrid statistical and rule-based incremental approach. InAsian Semantic Web Conference (pp. 420-433). Springer, Berlin, Heidelberg, Dec 2008

AUTHORS PROFILE

A.Shiny is currently a faculty advisor in SRM Institute of Science and Technology. Her current research interest is Network Security.

Reuma Akhtar is currently pursuing her final year in B.Tech (CSE) from SRM Institute of Science and Technology.

Saurav Singh is currently pursuing his final year in B.Tech (CSE) from SRM Institute of Science and Technology.

K. Sujana is currently pursuing her final year in B.Tech (CSE) from SRM Institute of Science and Technology.

D.V. Neelesh is currently pursuing his final year in B.Tech (CSE) from SRM Institute of Science and Technology.