IPIG: A Geospatial Analysis Framework for Internet Protocol Address Intelligence Gathering

Meshesha K. Cherie, Houssain Kettani*

The Beacom College of Computer & Cyber Sciences, Dakota State University, Madison, South Dakota, USA.

* Corresponding author. Email: Houssain.kettani@dsu.edu

Manuscript submitted November 23, 2018; accepted January 3, 2019.
doi: 10.17706/ijcce.2019.8.3.93-103

Abstract: This paper presents a unified geospatial analysis framework of Internet Protocol (IP) addresses for Intelligence Gathering (IG). Geospatial analysis is the exploitation and analysis of imagery and location-based activities on earth. Internet connected devices have IP addresses that can be resolved to a geographic location. The IP addresses have metadata which includes location, registry and network information, Autonomous System and Internet Service Provider. These IP address records are maintained by different service organizations. Using separate services to get an IP address information is not effective. Since data is dispersed in different datasets, integration work is important for detailed analysis. By compiling different services and applying integration and analysis techniques, this paper aims to provide a unified framework and approach that can be used by security analysts for the purpose of IP address intelligence gathering and network traffic analysis. The geospatial analysis framework proposed in this paper or IPIG, is composed of IP address search engine, geospatial computations, IP similarity measurement and geo mapping techniques. IPIG is designed to handle multiple datasets and services and is validated via a test application. Using IPIG provides an improved way of IP address intelligence gathering for both technical and non-technical users.

Key words: Geospatial intelligence, IP address, IP locator, GeoIP, IP similarity, IP search, network traffic analysis, intelligence gathering, mapping.

1. Introduction

Nowadays, there are over four billion IP addresses globally. Each IP address has associated records. Storing large number of IP addresses along with their associated attributes requires a large dataset. Therefore, IP addresses intelligence gathering is a resource intensive process. There are several IP address analysis tools such as IP to geo locator, IP Whois, interactive maps, blocked IP feed providers, etc., that are used by security analysists in case of incidents and intelligence gathering. These tools are maintained by different entities. For example, GeoIP databases are maintained by organizations such as Maxmind [1], while IP registry information is maintained by Internet Registry organizations such as American Registry for Internet Numbers (ARIN) [2] and risky IP address datasets are maintained by FireHOL [3]. If security analysts want to get complete information about an IP address such as location, registry information, reputation, risk, etc., they have to use these tools and services separately. For instance, a GeoIP database does not have IP registry information, Internet Registry organizations do not hold IP location or reputation information, IP reputation datasets hold a list of IP addresses with no additional information about the IP addresses, and so on. These IP analysis tools and services are stand-alone by themselves and function separately. Packaging them in one does not bring data coherency or provide cohesive analysis. Therefore, a special integration and analysis approach, which is the focus of this research, is important.

The purpose of this research is to design a unified framework that integrates several datasets and services by applying different techniques such as search engine, spatial search, geospatial computation, IP set similarity and geo mapping in order to provide robust IP address intelligence gathering and analysis service. A search engine can handle a large set of data and provides fast performance. In this paper, use of the search engine technology to index IP addresses with their associated records is discussed in detail. The IP address location data is handled using spatial search engine along with other IP address related records. The proposed framework or IPIG, handles search queries such as risky IP addresses in a specific location or within a specified radius of a given location, displaying IP address information belonging to a given autonomous system or organization. Geospatial analysis involves data visualization on a geographic map. Since IP addresses can be mapped to geographic locations, representing IP search and analysis results on a visual map communicates information in a clear and efficient way. The framework employs data mapping techniques to achieve this. The geo mapping techniques are generic in nature, which means, using this technique the analyst can visualize IP related data such as traceroute data on an interactive map.

In case of Distributed Denial of Service (DDoS) attacks, a security analyst may want to find out any similarity between the current attack and known previous attacks. The IP set similarity measure incorporated in the proposed framework measures substantial similarity between two sets of IP addresses with respect to the attributes of the IP addresses. Architecture of the proposed framework is designed to be multi-layered in order to decouple components from each other. This makes the components less dependent on specific data type or platform. A stand-alone validation application was developed to evaluate IPIG using publicly available datasets. Each technique implemented in IPIG is tested and validated in a lab project. In the next section we describe related work, while in Section 3 we describe IPIG in detail, and an illustrative example is discussed in Section 4. We finally present concluding remarks in Section 5.

2. Related Work

Mapping IP addresses to geographic locations is the first step in performing geospatial intelligence on IP addresses. Several techniques have been practiced to determine geographic locations of IP addresses. Accordingly, in [4] three methods to determine geographic location from an IP address were proposed. These are:

1. GeoTracking: Which extracts geographic information such as city, state or country from Domain Name System (DNS) of hosts and routers.
2. GeoPing: Which determines locations using network delay measurements from known locations, and
3. GeoCluster: Which determines geographic location from use of network routing by combining partial IP-to-location mapping with Broder Gateway Protocol (BGP) routing. This third approach is the best among the three approaches with error deviation between 28 kilometers and few hundred kilometers [4].

In the software industry, Google unveiled the use of mobile devices and laptops to determine geographic location of networked devices on the Internet [5], [6]. Other companies like Maxmind store and distribute GeoIP databases [1]. The IP address to geographic location mapping accuracy is crucial for the best analysis result. Accordingly, in [7] the accuracy of GeoIP location databases such as HostIP, IP2Location, InfoDB, Maxmind and Software77 was investigated and the results show that geolocation databases are effective in mapping IP to country. Although city-level IP mapping is effective for a few popular countries like the USA, it has significant errors for most countries.

Network visualization and analysis have been studied by different researchers and practitioners. Several lists of networks and internet data analysis tools such as geographic tools, security tools and topology tools can be found on the Center for Applied Internet Data Analysis (CAIDA) website and multiple traffic analysis research projects are being conducted on CAIDA platform [8]. Most research on CAIDA is based on the global internet backbone to provide insights into the macroscopic function of the Internet infrastructure, behavior, usage and evaluation. The global Internet as a whole is a complex network to visualize. A hierarchical network map is presented in [9] that shows a hierarchy of the seven continents, 190 countries, 23,054 autonomous systems and 197,427 IP address prefixes. The model shows the hierarchy on a rectangular grid using TreeMap algorithms. Some network visual analysis tasks such as traceroute visualization may not require the global internet visualization. The Traceroute tool was originally written by Van Jacobson in 1988 [10]. Traceroute does not provide geographic location information of nodes along the route. But the multithreaded graphical traceroute tool called GTrace displays IP address, node name, location as (longitude, latitude) and Round-Trip-Times (RTT) [10]. GTrace does not display details such as registry information or reputation of the IP addresses in the route and it is listed as an unsupported tool on the CAIDA portal [8].

Network traffic can be identified by predefined patterns, IP headers, protocols or ports. However, such predefined patterns may not actually determine the traffic type in some cases like DDoS attacks using random port numbers attacking a web server. For better understanding of network traffic, traffic flow characterization mechanisms are useful. In previous works, a method for traffic characterization was developed, which automatically classifies traffic into small clusters [11]. This traffic clustering algorithm can analyze multiple traffic dimensions at once such as source, destination. protocol, source port and destination port. Traffic characterization that relies on protocols could be inaccurate if the source IP is spoofed. The IP address location visualization would be also inaccurate if the IP address happens to be spoofed. A CAIDA Spoofer tool uses Source Address Validation (SAV) algorithm to detect IP spoofing using routing loops appearing in the traceroute data to infer the absence of filtering by a provider autonomous system at a provider-customer interconnect [12].

Open-source-based Internet data analysis framework, BGPStream is used for historical and real-time BGP data analysis [13]. The software framework enables other applications to process large amounts of distributed and live BGP measurement data for monitoring and investigation such as BGP hijacking attack detections and connectivity disruption detection. The purpose of BGP is to exchange reachability information for autonomous systems. The BGPStream team developed a data visualization tool based on Autonomous System Number (ASN). Although the BGPStream tool detects attacks, it does not provide IP address level details such as IP reputation. Interactive maps based on the global network such as the global real-time cyberthreat map [14] and submarine cable map [15] display a predefined query data that could be useful for overall assessment of the global Internet. Even though these maps are useful to determine the global cyberthreats, they lack processing particular threat related inquires because they provide predefined data. Geospatial analysis is useful for security analysists to track down malicious activities based on location information. On the other hand, if misused, it could be a tool for adversaries. Location data gathered from smart devices such as fitness devices and smart watches could reveal sensitive information. A good example of this is Strava's fitness tracker heat map that revealed in 2018 the locations of several undercover American military bases worldwide [16].

3. Geospatial Analysis Framework

This research proposes an application framework model that comprises geospatial methods that can be used for IP address Intelligence Gathering or IPIG. Generally, IPIG handles queries from users, processes them and displays results. An IP address search engine, geospatial commutations, IP similarity measurements and geo mapping techniques are the major feature components of the framework. Data processors and data access Application Programming Interfaces (API) are also included in the framework. The implementation architecture of IPIG is shown in Fig. 1. The framework has three layers: User interface, core logic processor and data access. A user query is parsed in the logic processor module, which in turn performs different activities such as IP address search, IP to Geo location mapping and geospatial computations. Processed results then return to the user interface module, which in turn depicts the results on a grid and an interactive map. The data processor module is responsible for periodic data update, indexing, and compiling the data for fast access and multi-threading support by applying different software design patterns. The framework is designed to work with both commercial and open-source datasets, but its effectiveness depends on the quality of the data it is using. In this paper, only publicly available datasets were used to validate the model. While IPIG is not restricted to support only specific features, in this paper we only discuss four major features. These are IP Search, IP Set Similarity, Distance Calculation and Mapping. Each of these features is described below.

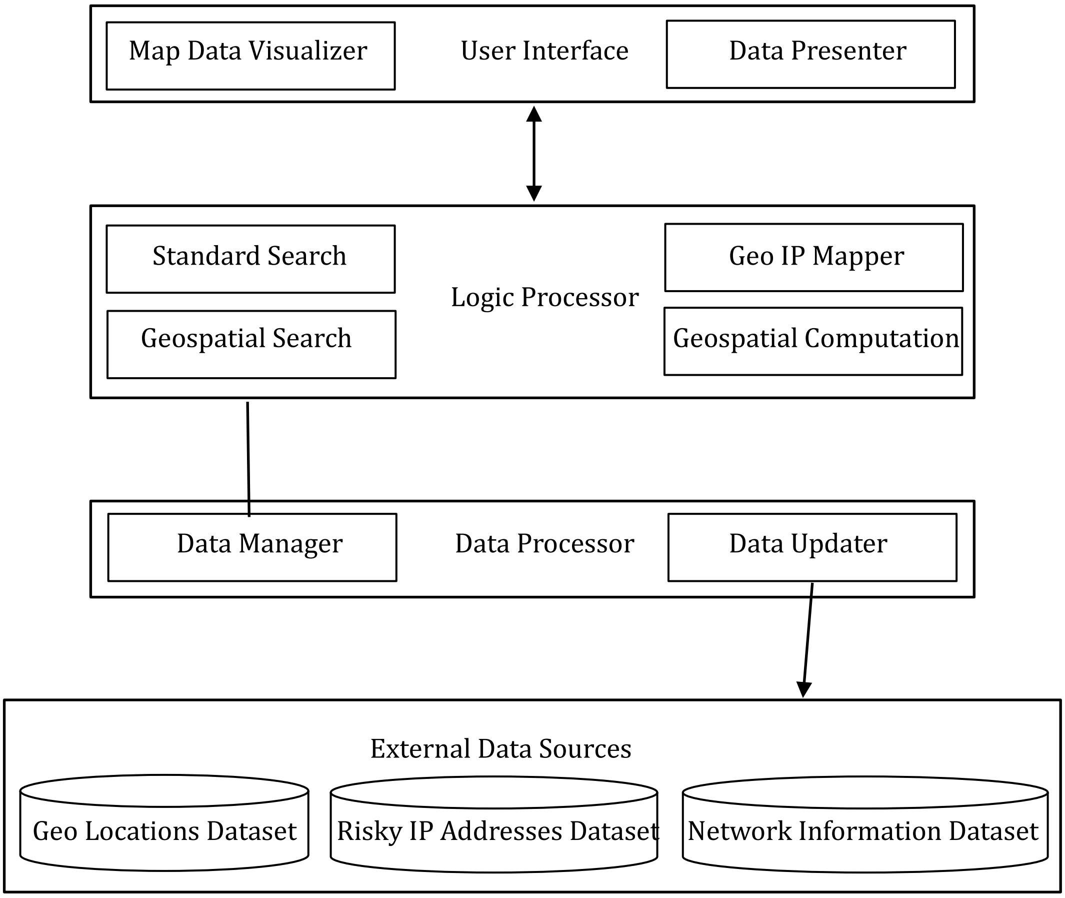


Fig. 1. IPIG application architecture.

3.1. IP Search

The IP address searching is the first step to perform before running geospatial analysis. As shown in Fig. 2, the Data Loader service searches GeoIP databases such as Maxmind, ARIN APIs, and IP reputation feeds to pull as much data as possible about the IP address in process and generates an IP detail object. In this process, an IP address gets converted to list of records. Due to the large number of IP addresses, the IP search is not a trivial algorithm, especially if the search involves IP address related records since IP related datasets are independently maintained, which complicates the search. IP registration is maintained by

Internet Registry organizations, Domain names are maintained by hosting companies while geographic information is maintained by other entities such as governments. For example, if the search requirement is to list down IP addresses associated with some domains or risky IP addresses in a given location. In this case, simple data search algorithms would be inefficient to handle such large and dispersed dataset. Therefore, the art of search engine technology is applied in IPIG.

Search engine implementation involves different phases; indexing and parsing are the major ones. The input data need to be indexed first. Data normalization and natural language processing tools can be applied for the best search result. As shown in Fig. 3, once the IP address is mapped to the IP detail object which holds different records, collection of words will be extracted from object property values and organized properly to form a text. The latter is then treated as a document with an IP address as a unique identifier in the search engine. Location coordinate data are analyzed as vectors and indexed as spatial data. Both the text and spatial data are combined to form an indexable document. Then each document that represents an IP address will be indexed by the search engine Indexer. After the indexing process is completed, data will be ready for searching. The Searcher component searches matching documents for the given query. For application users, searching would be similar to using search engines like Google or Bing. The main role of the search engine is to speed up searching for the IP address and related records. With fast searching capability, further analysis which is discussed next can easily be applied.

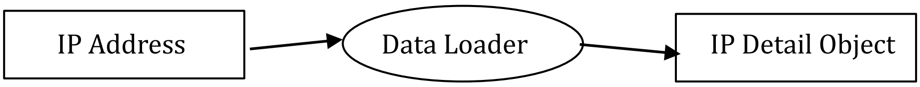


Fig. 2. IP address to IP detail object conversion.

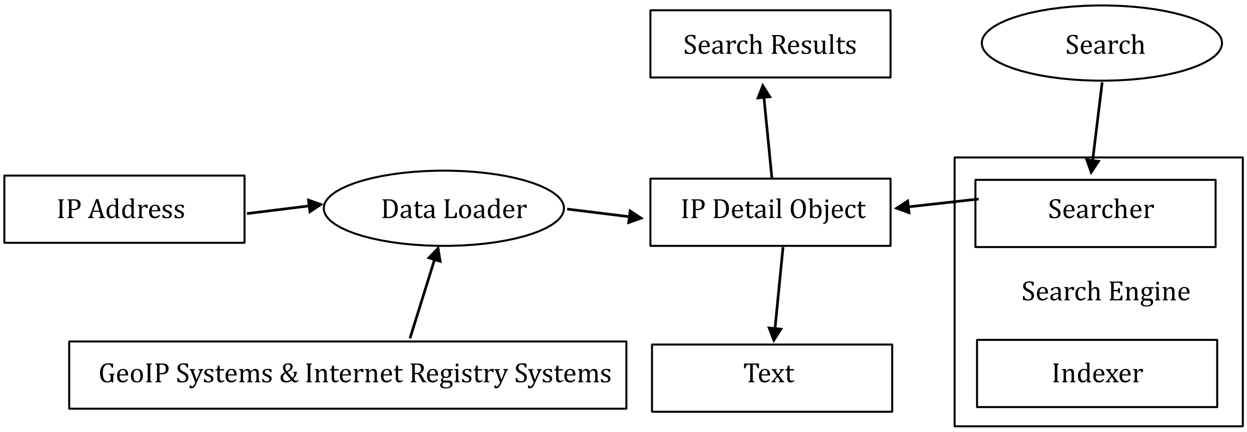


Fig. 3. High-level architecture of IP details search engine.

3.2. IP Set Similarity

Previous work shows that IP address similarity can be measured using Word-to-Vector text data mining technique [17]. This data mining technique converts IP to vector and captures similarity of IP addresses based on their network communication. In this paper, IP Set Similarity determines the substantial similarity between two IP sets based on the attributes of the IP addresses by taking advantage of the IP search feature discussed earlier. For instance, a security analyst may want to determine any similarity between DDoS attacks by comparing log files containing IP addresses. The IP Set Similarity identifies similarity between two sets of IP addresses by applying set similarity measures on the attributes of each IP address. Attributes may include location, network information, or Internet Service Provider (ISP) information. Algorithm 1 shows how to measure similarity between two given IP sets: *ip_set1* and *ip_set2*.

Algorithm 1: IP Set Similarity Measurement

```
Step-1: Select comparison attributes.
comparison_attrs ← (Country, Region, City, ASN, ISP, Network)
Step-2: For each IP sets, apply IP search to map each IP address to IP detail object
ipdetails_list1 ← ip_set1.Map(ip_detail ← IPSearchResult(ip))
ipdetails_list2 ← ip_set2.Map(ip_detail ← IPSearchResult(ip))
Step-3: For each comparison attribute, generate distinct set of attributes for both ipdetails_list1 and ipdetails_lsit2. Then run set similarity measures on each pair of distinct attribute set.
    for each attr in comparison_attrs
        attr_set1 ← ipdetails_list1.Select(i ← i.attr).Distinct()
        attr_set2 ← ipdetails_list2.Select(i ← i.attr).Distinct()
        sim_attr ← Similarity(attr_set1, attr_set2)
    end for
```

Set similarity measures such as Euclidean Distance, Manhattan Distance and Cosine Similarity determine similarity between two sets when elements can be represented as points or vectors. On the other hand, Jaccard Index and Overlap coefficient can be used when elements are objects [18]. Geographic coordinate values of IP addresses can be treated as vectors and similarity measures can be computed on the vector elements. However, the coordinate values are already mapped to named objects such as cities and regions in the IP search engine. Therefore, for high-level network traffic analysis, Jaccard Index and Overlap coefficient similarity measures are sufficient to measure similarity between two IP sets representing network traffic. Jaccard index can be used to measure the overall similarity of the two IP sets on each attribute. The Jaccard Index is a percentage of the number of common objects in two sets out of the total objects in the sets. Accordingly, the Jaccard similarity of two sets A and B can be expressed as: J(A, B) = |A∩B| / |A∪B|. Overlap coefficient also known as Szymkiewicz–Simpson coefficient, is similar to Jaccard Index but it is the measure of intersection divided by the smaller size of the two sets: Overlap(A, B) = |A∩B| / min(|A|, |B|).

3.3. Distance Calculation

Once IP addresses are mapped to geographic locations and coordinate systems, the distance between two IP addresses can be calculated using the Great Circle or Orthodromic Distance formula. The latter uses Haversine Formula which ignores elevation differences and presumes the spherical earth with constant radius R, which is approximately 6,367 kilometers or 3,956 miles [19]. For example, distance (d) between two locations A(lon_a, lat_a) and B(lon_b, lat_b) can be calculated as $d = Rc$, where

$$c = 2\sin^{-1}(\min(1, a)) \ ,$$

$$a = \sqrt{\sin^2(\Delta lat/2) + \cos(lat_a)\cos(lat_b)\sin^2(\Delta lon/2)} \ ,$$

$$\Delta lon = lon_b - lon_a \text{ and}$$

$$\Delta lat = lat_b - lat_a.$$

3.4. Mapping

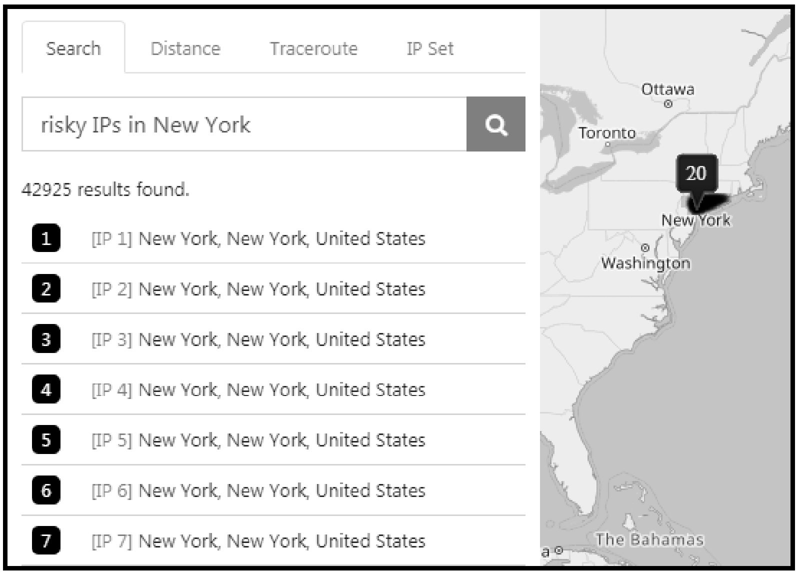
Data visualization provides efficient communication to the user. The proposed IPIG has a geo mapping technique in which IP search and analysis results can be visualized on a geographic map. This capability enables security analysts to visualize a given set of IP addresses on a geographic map along with the corresponding details.

4. Illustrative Example

This section presents examples and illustrations of a validation application that uses the proposed geospatial analysis framework IPIG. Examples of the IP Search, IP Sets Similarity, distance between two IP addresses, and mapping features are discussed in this section.

4.1. Geospatial Search

Searching IP addresses by vicinity or location may be required in some cases. For example, a user may want to search for a list of risky IP addresses in a given location before connecting to a public network. The location could be a city, region or country. Fig. 4 shows risky IP addresses in New York City, where each IP row in the grid can be expanded to show the details of the selected IP address such as network registry information, owner of the IP address, etc. A large number of IP addresses may be listed as a search result. The search engine technique has a fast response time. In order to evaluate the search speed, queries that bring a large number of records such as risky IP addresses in a given country is used. Table 1 shows the performance of the IP search engine while searching for risky IP addresses for selected countries shown in the Table.



risky IPs in New York

42925 results found.

1 [IP 1] New York, New York, United States

2 [IP 2] New York, New York, United States

3 [IP 3] New York, New York, United States

4 [IP 4] New York, New York, United States

5 [IP 5] New York, New York, United States

6 [IP 6] New York, New York, United States

7 [IP 7] New York, New York, United States

Fig. 4. Risky IP addresses in New York City.

Table 1. IP Search Engine Performance

Country	Risk IP Addresses Count	IP Search Elapsed Time
Canada	13,173	386 ms
China	212,021	79 ms
Japan	8,049	81 ms
Russia	114,188	496 ms
USA	269,393	601 ms

4.2. IP Set Similarity Measurement

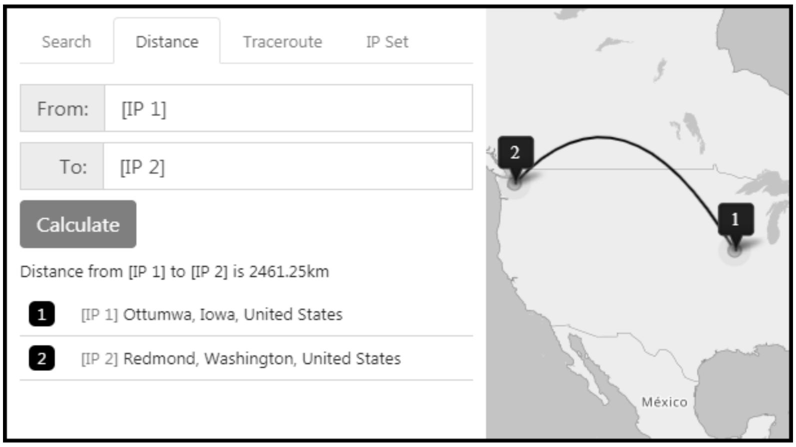
One advantage of the IP address searching is to quickly map a large set of IP addresses to a set of IP details. This conversion is useful when a security analyst wants to know any substantial similarity between two sets of IP addresses. For example, a security analyst may want to identify similarities between two attacks by comparing corresponding log files containing IP addresses. The proposed framework has a log parser that can extract IP addresses from a given log file. Table 2 shows similarity indices measured between two IP sets extracted from two log files. The first log file has over 90,000 Hyper-Text Transfer Protocol (HTTP) requests and the other log file has over 100,000 HTTP requests. Both log files are taken from daily traffic of a website. As shown in the Table 2, 84% of the traffic has country overlap. The closer the values of the Jaccard Index and Overlap Coefficient to one, the more the two sets are similar.

Table 2. Similarity Measure between Two IP Sets

Comparison Attribute	Jaccard Index	Overlap Coefficient	Intersection Count
Country	0.68	0.84	61
Region	0.39	0.59	167
City	0.28	0.45	283
ASN	0.34	0.56	208
ISP	0.33	0.56	195

4.3. Distance between Two IP Addresses

In some cases, it may be required to determine the geographic distance between two IP addresses such as the distance between two data servers. Once each IP address is mapped to a geographic coordinate using the IP search engine, the Great Circle or Orthodromic Distance formula is used to compute the distance between the two coordinates. Note that this distance is not the actual distance that a packet travels from source to destination server but rather it is an air distance between the two locations. For example, Fig. 5 shows the distance between two IP addresses on the map.



From: [IP 1]

To: [IP 2]

Distance from [IP 1] to [IP 2] is 2461.25km

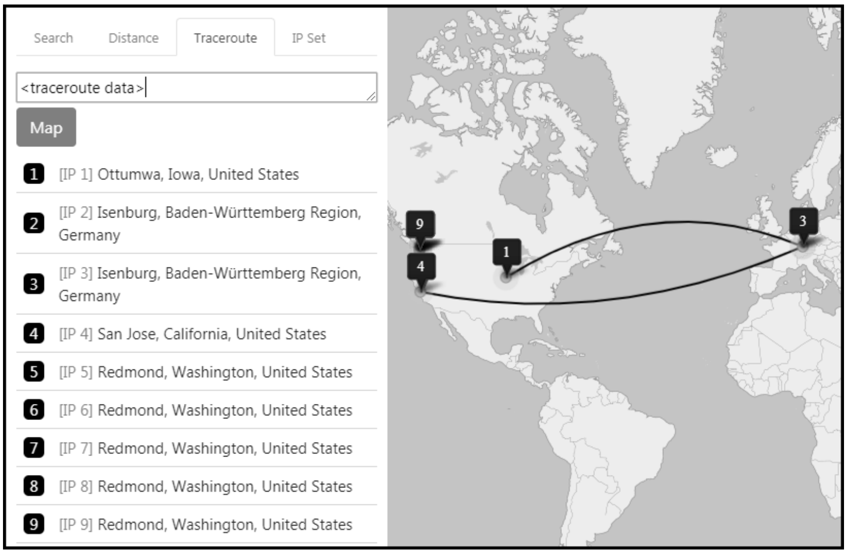
1 [IP 1] Ottumwa, Iowa, United States

2 [IP 2] Redmond, Washington, United States

Fig. 5. Distance between two IP addresses.

4.4. Mapping

Locating IP addresses on a geographic map enables security analysts to visualize IP address related information such as distribution of risky IP addresses in a given location or packet trace routing. For example, Fig. 6 shows the geo mapping component in IPIG displaying details of a traceroute data.



<traceroute data>

1 [IP 1] Ottumwa, Iowa, United States

2 [IP 2] Isenburg, Baden-Württemberg Region, Germany

3 [IP 3] Isenburg, Baden-Württemberg Region, Germany

4 [IP 4] San Jose, California, United States

5 [IP 5] Redmond, Washington, United States

6 [IP 6] Redmond, Washington, United States

7 [IP 7] Redmond, Washington, United States

8 [IP 8] Redmond, Washington, United States

9 [IP 9] Redmond, Washington, United States

Fig. 6. Traceroute map.

5. Concluding Remarks

This paper has provided a unified geospatial analysis framework which integrates several functionalities such as IP detail search, IP similarity measures, and GeoIP mapping by applying search engine, analysis and mapping techniques. The proposed framework or IPIG, has been evaluated by a stand-alone validation application. The proposed framework can be used in software applications. With these features, using IPIG as stand-alone or integrating it with other applications, simplify and improve IP address intelligence gathering tasks. Applications, with IPIG can identify risky IP addresses in a given location. Therefore, in addition to technical users, others could also be benefited from such applications to prevent their systems from malicious connections. Although IPIG is designed to work with different datasets, data access APIs need to be added in order to support additional datasets which could be GeoIP database, risky IP dataset, geo name database, etc. For instance, geo name databases from geonames.org and nga.mil could increase accuracy of the IP to geo location mapping. A special Query Analyzer in the context of network traffic analysis could also optimize the IP searching feature. Finally, IPIG can be enhanced to a web application to provide service to the public.

References

[1] Maxmind. (2018). *GeoIP2 .NET API.* Retrieved from https://maxmind.github.io/GeoIP2-dotnet/

[2] American Registry for Internet Numbers (ARIN). (2018). *ARIN at a glance.* Retrieved from https://www.arin.net/about_us/overview.html

[3] FireHOL. (2018). All cybercrime IP feeds. *FireHOL.* Retrieved from http://iplists.firehol.org/

[4] Padmanabhan, V. N., & Subramanian, L. (2001). An investigation of geographic mapping techniques for internet hosts. *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'01)* (pp. 173-185). New York, NY: ACM. Retrieved from https://doi.org/10.1145/383059.383073

[5] Goeller, K., Spaulding, B., Godwin, J. P., Anderson, B., & Le-Chau, L. (2007). U.S. Patent No. 7,200,658 B2. Washington, DC: U.S. Patent and Trademark Office.

[6] Preston, D., & Preston, J. (2001). U.S. Patent No. 6,236,652 B1. Washington, DC: U.S. Patent and Trademark Office.

[7] Poese, I., Uhlig, S., Kaafar, M. A., Donnet, B., & Gueye, B. (2011). IP geolocation databases: Unreliable?

ACM SIGCOMM Computer Communication Review, 41, 53-56. Retrieved from https://doi.org/10.1145/1971162.1971171

[8] Center for Applied Internet Data Analysis (CAIDA) Tools. (2018). *Overview of CAIDA Software Tools*. Retrieved from http://www.caida.org/tools/

[9] Mansmann, F., Keim, D. A., North, S. C., Rexroad, B., & Sheleheda, D. (2007). Visual analysis of network traffic for resource planning, interactive monitoring, and interpretation of security threats. *IEEE Transactions on Visualization and Computer Graphics*, 1105-1112. Retrieved from https://doi.org/10.1109/TVCG.2007.70522

[10] Periakaruppan, R., & Nemeth, E. (1999). GTrace - A graphical traceroute tool. *Proceedings of the 13th Systems Administration Conference (LISA'99)* (pp. 68-78). Berkeley, CA: The USENIX Association.

[11] Estan, C., Savage, S., & Varghese, G. (2003). Automatically inferring patterns of resource consumption in network traffic. *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '03)* (pp. 137-148). New York, NY: ACM. Retrieved from https://doi.org/10.1145/863955.863972

[12] Lone, Q., Luckie, M., Korczyński, M., & Eeten, M. V. (2017). Using loops observed in traceroute to infer the ability to spoof. *Proceedings of the 18th International Conference on Passive and Active Measurement (PAM 2017)* (pp. 229-241). *Lecture Notes in Computer Science*. Heidelberg: Springer. Retrieved from https://doi.org/10.1007/978-3-319-54328-4_17

[13] Orsini, C., King, A., Giordano, D., Giotsas, V., & Dainotti, A. (2016). BGPStream: A software framework for live and historical BGP data analysis. *Proceedings of the 2016 Internet Measurement Conference (IMC'16)* (pp. 429-444). New York, NY: ACM. Retrieved from https://doi.org/10.1145/2987443.2987482

[14] Kaspersky Lab. (2018). *Cyberthreat real-time map.* Retrieved from https://cybermap.kaspersky.com/

[15] PriMetrica Inc. (2018). *Submarine cable map*. Retrieved from https://www.submarinecablemap.com/

[16] Hsu, J. (2018, January 29). The Strava heat map and the end of secrets. *Wired.* Retrieved from https://www.wired.com/story/strava-heat-map-military-bases-fitness-trackers-privacy/

[17] Ring, M., Landes, D., Dallmann, A., & Hotho, A. (2017). IP2Vec: Learning similarities between IP addresses. *Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 657-666). Piscataway, NJ: IEEE. Retrieved from https://doi.org/10.1109/ICDMW.2017.93

[18] Vijaymeena, M. K., & Kavitha, K. (2016). A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal (MLAIJ), 3*, 19-28. Retrieved from https://doi.org/10.5121/mlaij.2016.3103

[19] Shumaker, B. P., & Sinnott, R. W. (1984). Astronomical computing: 1. Computing under the open sky. 2. Virtues of the haversine. *Sky & Telescope, 68*, 158-159**.**

Meshesha Kibret Cherie received the B.S. degree in computer science from Addis Ababa University, Ethiopia, in 2005 and the M.S. degree in computer science from Maharishi University of Management, Fairfield, IA, in 2009 and is currently pursuing his Ph.D. in cyber operations from Dakota State University. He has been the chief technology officer of Kekros Systems since 2012, which is a software development and technology consulting company based in Fairfield, IA, specializing in Microsoft platform solutions, mobility development, artificial Intelligence, geospatial intelligence, and custom software and components development.

Houssain Kettani received the B.S. degree in electrical and electronic engineering from Eastern Mediterranean University, Cyprus in 1998, and the M.S. and Ph.D. degrees both in electrical engineering from the University of Wisconsin at Madison in 2000 and 2002, respectively. Dr. Kettani served as a faculty member at the University of South Alabama (2002-2003), Jackson State University (2003-2007), Polytechnic University of Puerto Rico (2007-2012), Fort Hays State University (2012-2016), Florida Polytechnic University (2016-2018) and Dakota State University since 2018. Dr. Kettani has served as a staff research assistant at Los Alamos National Laboratory in summer of 2000, a visiting research professor at Oak Ridge National Laboratory in summers of 2005 to 2011, a visiting research professor at the Arctic Region Supercomputing Center at the University of Alaska in summer of 2008 and a visiting professor at the Joint Institute for Computational Sciences at the University of Tennessee at Knoxville in summer of 2010. Dr. Kettani's research interests include computational science and engineering, high performance computing algorithms, information retrieval, network traffic characterization, number theory, robust control and optimization, and Muslim population studies. He presented his research in over seventy refereed conference and journal publications and his work received over five hundred citations by researchers all over the world. He chaired over hundred international conferences throughout the world and successfully secured external funding in millions of dollars for research and education from US federal agencies such as NSF, DOE, DOD, and NRC.