Malicious Website Detection Using Bloom Filter

Prof.Spurti Shinde[1], Iqra Khan[2] ,Kaustubh Jagtap[3], Shreyas Dashpute[4], Madhur Kharche[5]
[1, 2, 3, 4, 5]Pimpri Chinchwad College Of Engineering

Abstract- *Bloomfilter is a kind of probabilistic type of data structure that gives an idea, promptly and memory efficientlyof the probability of an element present in a set or not. It is a simple binary data structure that helps in testing a set of membership. Bloom filter is probabilistic in nature that means it may give false positive value but would never give a false negative value. This paper introduces the idea of implementation of bloom filter to protect from the threat of malicious websites and improve cache performance. This idea basically blooms from the process of hashing. Ideas other than this is also explored during the applications. This concept is promising in the future as whole world is heading towards digital age. Bloom filter helps in uplifting the standard of web browsing and reduce the threat from malicious sites which is a concern over the globe.*

Keywords- Malicious websites, Privacy-Preserving Medical Data Sharing,Secure Cloud Storage Service,Bloom Filter algorithm,Error Detection, Hashing, false positive value, false negative value.

I. INTRODUCTION

With the ongoing speed development there is a huge need of data to be searched. Meanwhile, people have to monitor and control a large amount of data, which increases management cost as well as there is loss of efficiency [1]. This may lead to the development of malicious website. Malicious website are websites which causes harm to the host device. They install malicious software on the device which further extracts data from the device or uses that device to enter into another network. Other work of these websites is to provide fake information or debatable news, information.

This is the need of the hour to prevent such malicious website from causing digital harm. But for normal being it is difficult as well as exhausting to detect that whether a website is malicious or not. To serve this purpose we need a kind of filter which on its own detects malicious website and help saving the valuable information. For this purpose there's implementation of bloom filter.

Bloom filter decides the presence of an element in a set.Till date there are many projects which have utilized bloom filter in many fields maybe as a translation function or for similarity check. Most of the time bloom filter is associated with other websites or applications to secure their users data.

In this paper we have proposed the scheme of detecting malicious website using bloom filter. Providing a database of list of malicious website it will search through the strings and return a true value for non-malicious website whereas a false value of a malicious website. Our scheme mostly utilizes the hashing concept for the verification purpose. Our scheme will provide an updated way to detect malicious website and will reduce the threat of malware installation which would eventually reduce the rate of cybercrime.

II. LITERATURE SURVEY

Bloom-Filter can be implemented with browsers for collecting and maintaining the data and information related to the web-cache based on the searches performed by the user [2].Bloom Filter can be applied as a browser extension so that when a user performs a search the searched or entered URL will be passed through the Bloom Filter, and is checked with the data set of the potentially harmful websites. Although there are chances of a False Positive, these chances are very less as the accuracy of the Bloom Filter is very good [3].Now-a-days since everyone uses Internet, it becomes a very important task to keep Adult Content containing or Pornographic Websites away from the youngsters. This can be brought through application of a Bloom Filter which can detect and warn the user if he/she searches for a website containing inappropriate content [4]. Due to the Denial of Service (DoS) Attacks, the processing of the system gets too overloaded as the targeted system is flooded with traffic which makes the system unable to respond and the system crashes. DoS Attacks makes the machine unusable for its Intended users. Such attacks can be used to attack the systems of the Hackers which will make their system unusable [5].Bloom filters are used to check, optimize or rectify the errors in the currently initialized data elements. This detection of errors and correction can be easily and efficiently be performed with the use of Bloom Filter [6]. RFID Tags (Radio Frequency Identification) are a kind of system used for tracking of items

by means of smart and sharp barcodes, which is distinct for each item. This barcode system makes it very easy to maintain data for each item. If any Unknown RFID Tag which potentially harmful, is detected by the Bloom Filter, a warning can be given so that any kind of harm can be avoided [7]. RFID (Radio Frequency Identification) Tags is a very key technology brought by Internet of Things (IoT). RFID Tags has numerous applications in various scenarios such as in tracking systems,database of registered vehicles,etc. These tags must be monitored periodically, this Object Monitoring can be carried out easily with the help of a Bloom Filter and any missing or unknown RFID tag can be detected easily [8]. Bloom Filter used in this paper is constructed using two bloom filters. The inevitable results produced from primary are removed by secondary bloom filter [9]. Quantum bloom filters and corresponding algorithms are created in order to add new elements as well as remove existing elements from database. This is a big advantage of quantum bloom filter [10]. Bloom filters are used in smart medical data centres in order to protect the privacy of customer's personal medical problems. Hence, bloom filter is very useful in maintaining privacy [11].Cloud is not trusted as it is not owned by a single person. Hence, the data in memory should be verified and insufficient data should be erased. This problem can be solved by doing data verification on IoT device itself. This idea is given in this paper [12]. Bloom filters have been very useful in networking as they enable the high speed, low-cost usage of various algorithms. This paper gives the idea of variable length signatures opposite to the practice of fixed length signatures
Bloom Filters can be used in detection of Harmful and Malicious websites, programs or applications. Another application for Bloom Filter can be detection of Porn0graphic or Adult Content containing sites, so in today's modern world where every age group is using internet, it can be used in Parental Controls which will keep disturbing and inappropriate content away from the children. Bloom Filters can also be used in designing efficient algorithms for extracting a uniform sample from the set of elements stored in the Bloom Filter. The bloom Filter also allows to modify the set very easily. Thus, it is evident that Bloom Filters find numerous applications in Security Applications.

III. EXPERIMENTATION AND RESULTS

3.1 Working Model:

Bit vector is the basic data structure of BLOOM FILTER. BLOOM FILTER is based upon the concept of Hashing. The process of Hashing includes use of an algorithm to convert any input into a fixed size text string. This means with the help of an algorithm the text isconverted in an array which is of letters and numbers. The data which is to be hashed is said to be input of BLOOM FILTER.. The output displayed is said to be HASH VALUE. Every HASH VALUE is unique. Hence different inputs result into different outputs. Therefore, same hash value should be produced by the same message. Hence Bloom Filter will process the input to create a unique hash value.

Hash Function:

A mathematical function used to convert an input into an integer is called Hash Function. The message which is to be hashed is called input of BLOOM FILTER. Hash function is the algorithm used to do the job. The output displayed is called as HASH VALUE. Every HASH VALUE is unique.

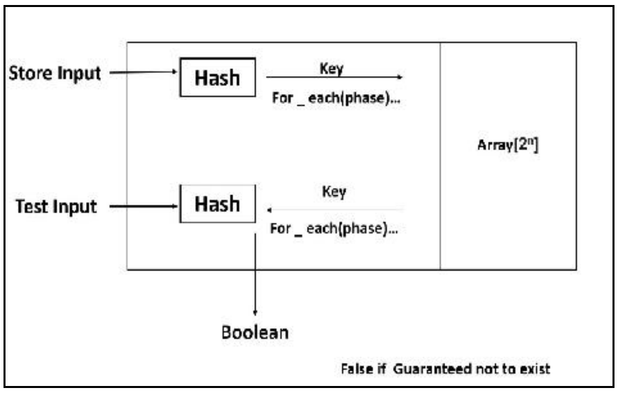


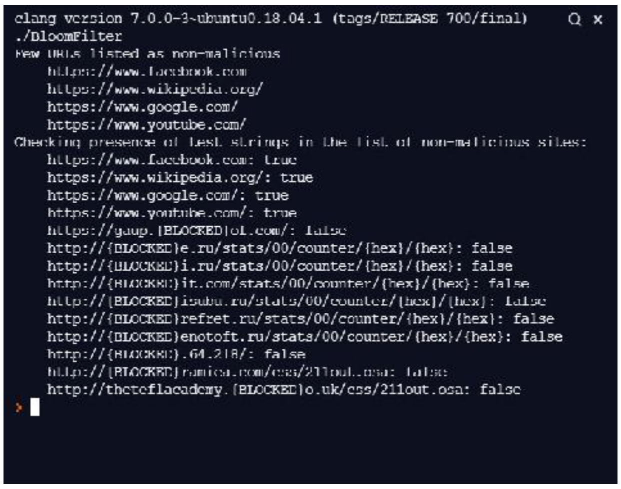
Fig1.1 Model of Bloom Filter

Bit Vector is the base data structure of bloom filter.The Bloom Filter is based upon the concept of Hashing. Hashing converts a given data as input of any length in a string which is of fixed size with the help of mathematically driven functions, which means the given text of any length, is converted into an array of numbers and letters through an algorithm. The message to be hashed is termed as input. This process is called hashing which is done with the help of hash function. The output is termed as Hash Value. Each output is different i.e. unique with respect to other. This means that it should not be possible to produce same hash value entering different inputs. Therefore, same input gives the same output i.e. hash value. Hence Bloom Filter will process the input to create a unique hash value.

Bloom Filter may give us false positive as well as false negative value. False positives is a binary test where system accepts the input as positive result though, it is negative result by certain similarities between positive and negative results. As entries in bloom filter increases, count of cases of false positive increases. It is usually called type-I error. E.g. Airport security: Items like coins or keys get recognized by system as metallic weapons. False negative is a binary test where system accepts the input as negative result though, it is positive result. It is usually called type-II error. E.g. Software testing: Search for virus by anti-virus system fails to catch the virus or malware.

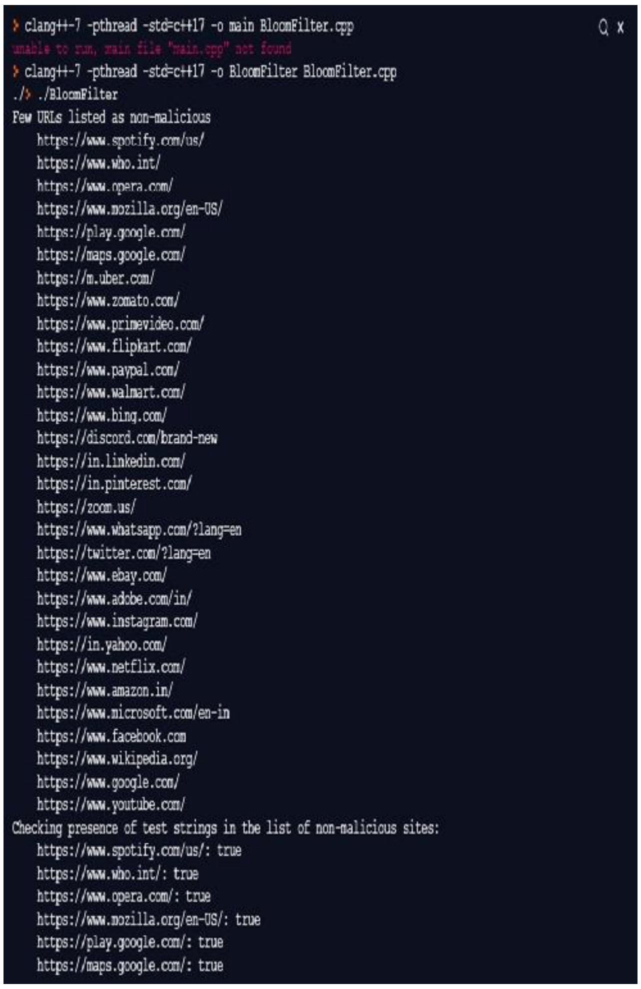
3.2. Results**:**

Following are the outputs of the results we get from the functioning of the provided bloom filter project which has detected the malicious websites from the given input.



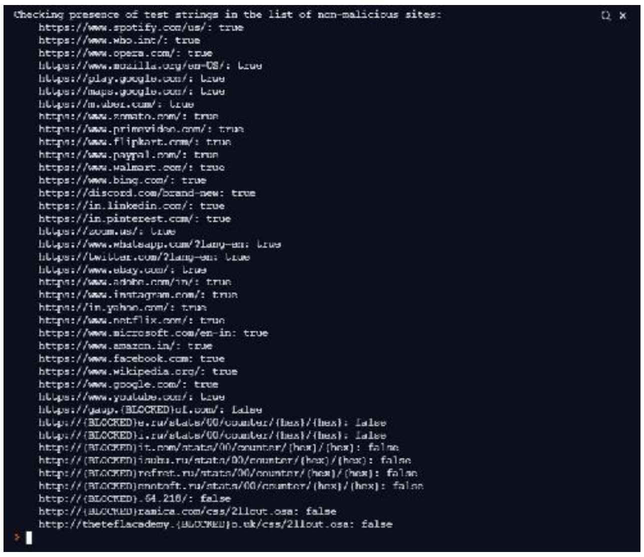
```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE 700/final)
./BloomFilter
Few URLs listed as non-malicious
    https://www.facebook.com
    https://www.wikipedia.org/
    https://www.google.com/
    https://www.youtube.com/
Checking presence of test strings in the list of non-malicious sites:
    https://www.facebook.com: true
    https://www.wikipedia.org/: true
    https://www.google.com/: true
    https://www.youtube.com/: true
    https://gaup.{BLOCKED}ol.com/: false
    http://{BLOCKED}e.ru/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}i.ru/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}it.com/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}isubu.ru/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}refret.ru/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}enotoft.ru/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}.64.218/: false
    http://{BLOCKED}ramica.com/css/211out.osa: false
    http://theteflacademy.{BLOCKED}o.uk/css/211out.osa: false
> 
```

Fig1.2 Result output 1



```
> clang++-7 -pthread -std=c++17 -o main BloomFilter.cpp
unable to run, main file "main.cpp" not found
> clang++-7 -pthread -std=c++17 -o BloomFilter BloomFilter.cpp
./> ./BloomFilter
Few URLs listed as non-malicious
    https://www.spotify.com/us/
    https://www.who.int/
    https://www.opera.com/
    https://www.mozilla.org/en-US/
    https://play.google.com/
    https://maps.google.com/
    https://m.uber.com/
    https://www.zomato.com/
    https://www.primevideo.com/
    https://www.flipkart.com/
    https://www.paypal.com/
    https://www.walmart.com/
    https://www.bing.com/
    https://discord.com/brand-new
    https://in.linkedin.com/
    https://in.pinterest.com/
    https://zoom.us/
    https://www.whatsapp.com/?lang=en
    https://twitter.com/?lang=en
    https://www.ebay.com/
    https://www.adobe.com/in/
    https://www.instagram.com/
    https://in.yahoo.com/
    https://www.netflix.com/
    https://www.amazon.in/
    https://www.microsoft.com/en-in
    https://www.facebook.com
    https://www.wikipedia.org/
    https://www.google.com/
    https://www.youtube.com/
Checking presence of test strings in the list of non-malicious sites:
    https://www.spotify.com/us/: true
    https://www.who.int/: true
    https://www.opera.com/: true
    https://www.mozilla.org/en-US/: true
    https://play.google.com/: true
    https://maps.google.com/: true
```

Fig1.3 Result output 2



```
Checking presence of test strings in the list of non-malicious sites:
    https://www.spotify.com/us/: true
    https://www.who.int/: true
    https://www.opera.com/: true
    https://www.mozilla.org/en-US/: true
    https://play.google.com/: true
    https://maps.google.com/: true
    https://m.uber.com/: true
    https://www.zomato.com/: true
    https://www.primevideo.com/: true
    https://www.flipkart.com/: true
    https://www.paypal.com/: true
    https://www.walmart.com/: true
    https://www.bing.com/: true
    https://discord.com/brand-new: true
    https://in.linkedin.com/: true
    https://in.pinterest.com/: true
    https://zoom.us/: true
    https://www.whatsapp.com/?lang=en: true
    https://twitter.com/?lang=en: true
    https://www.ebay.com/: true
    https://www.adobe.com/in/: true
    https://www.instagram.com/: true
    https://in.yahoo.com/: true
    https://www.netflix.com/: true
    https://www.microsoft.com/en-in: true
    https://www.amazon.in/: true
    https://www.facebook.com: true
    https://www.wikipedia.org/: true
    https://www.google.com/: true
    https://www.youtube.com/: true
    https://gaup.{BLOCKED}ol.com/: false
    http://{BLOCKED}e.ru/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}i.ru/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}it.com/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}isubu.ru/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}refret.ru/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}enotoft.ru/stats/00/counter/{hex}/{hex}: false
    http://{BLOCKED}.64.218/: false
    http://{BLOCKED}ramica.com/css/211out.osa: false
    http://theteflacademy.{BLOCKED}o.uk/css/211out.osa: false
> 
```

Fig1.4 Result output 3

3.3. Pie Chart**:**

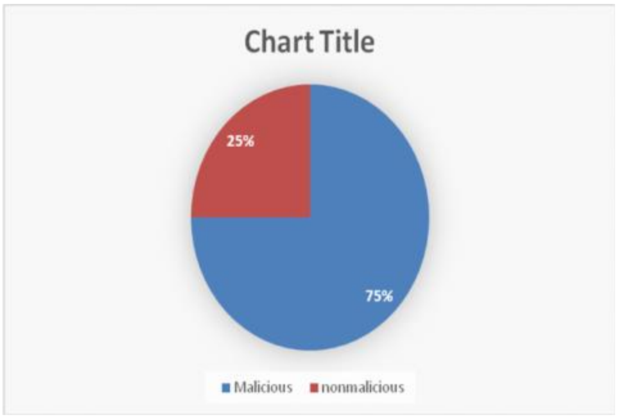


Fig1.5 Pictorial representation of website detection.

The Bloom Filter checked 40 websites. Of these 10 turned out to be malicious sites while 30 were non-malicious.

As per the fig1.2 of result pasted, The following URLs are set as few Examples of Non-Malicious (i.e. Are Safe for the Users to proceed Accessing and using those URLs):

1. https://www.facebook.com/
2. https://www.wikipedia.com/
3. https://www.google.com/
4. https://www.youtube.com/

The above URL are passed as strings through the Hash Function and the bit values are initialized in the vector on the basis of processing of the string passed through hash functions. When user enters one of these URLs the result is given as True, which means the URL is Non-Malicious and safe to proceed.

This evaluation is done by taking the entered URL by the user as a string and passing through the same Hash Functions used for initialising the Non-Malicious sites database.

When the user enters Any Other URL other than the initialised ones, the result is given as False, Which means that the entered URL might be Malicious and thus are not safe for accessing and it is better for the user to avoid using those websites.

For example , in Fig 1.2, if we consider https://www.google.com URL : Since Google is one the most trusted sources when it comes to internet and is used on a very large scale, the website is marked as Non-Malicious and is safe for the users to Access the site. To initialize the URL as non-malicious, the URL is first passed through the Bloom Filter and is treated as a string and a particular defined process is applied on the string with help of the Hash Functions and as per the output given by the Hash Functions , the bits are set in the Bloom Filter.

Now when the user searches for https://www.google.com , the URL is passed through the same Hash Functions which were also used for initializing the Non-malicious websites data set. After passing through bloom filter, the output returned after process performed by the Hash Functions on the searched or entered URL, this Output is compared with the bits of the data set of the Non-Malicious websites data set. If the bit values of searched URL matches with those of the initialized data set, the bloom filter returns ‘True’ as result i.e. the searched URLmatches the data set of the Non-malicious websites data set which means the searched URL is non-malicious and thus the user is allowed to proceed further with the website without any warning.

And if in case, the bit values of the output returned by the hash functions after processing the entered URL do not match the initialized data set of Non-Malicious websites, the result will be returned as ‘False’, which will mean that the Searched URL is not present in the list of websites marked as Non-Malicious i.e. it may be Harmful for the user to Access that website. If the website is malicious, a warning can be displayed on the screen of the user warning him/her about the website being potentially harmful to visit and suggest the user to Use an alternative website.

Since the bloom filter sets data as bits in a vector, It makes storing the data very storage efficient as large number of strings can be stored in a very small amount of Storage.

IV. OBSERVATIONS

- Very storage efficient
- Although there is scope for ‘False Positive’ , the Accuracy of the Bloom Filter still very good
- As we increase the use of Hash Function, the Entered and Initialised Data is passed through a large numbers of Hash Functions where each hash function will set value of 1 bit. So by increasing the use of bits available andhash functions, we are able to increase efficiency to some extent

This Bloom filter can be implemented as a browser extension, using which the searches performed by the user will Be first checked and if the user is accessing a Potentially Malicious website, then a message will be displayed ,warning the user About avoiding to proceed further will the entered URL.

V. CONCLUSION AND FUTURE SCOPE

We propose a safe and efficient search scheme regarding malicious website. Here the bloom filter will verify check for each website with its provided database and will return a true value for non-malicious and a false value for a malicious website. As the world is heading towards digital age the protection from cyber threat is a must which will be provided with bloom filter and such applications. It will not only secure our valuable information but will also help in getting more precise and veracious information. Finally, the resultant output showed that the proposed scheme will provide a more relevant way to detect malicious websites.

REFERENCES

[1] C. Guo, R. Zhuang, C. Chang and Q. Yuan, "Dynamic Multi-Keyword Ranked Search Based on Bloom Filter Over Encrypted Cloud Data," in IEEE Access, vol. 7, pp. 35826-35837, 2019, doi: 10.1109/ACCESS.2019.2904763.

[2] C. Jing, "Application and Research on Weighted Bloom Filter and Bloom Filter in Web Cache," 2009 Second Pacific-Asia Conference on Web Mining and Web-based Application, 2009, pp. 187-191, doi: 10.1109/WMWA.2009.51.

[3] K. Nandhini and R. Balasubramaniam, "Malicious Website Detection Using Probabilistic Data Structure Bloom Filter," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019, pp. 311-316, doi: 10.1109/ICCMC.2019.8819818.

[4] H. Yu, R. Cong, L. Chen and Z. Lei, "Blocking pornographic, illegal websites by internet host domain using FPGA and Bloom Filter," 2010 2nd IEEE InternationalConference on Network Infrastructure and Digital Content, 2010, pp. 619-623, doi: 10.1109/ICNIDC.2010.5657855

[5]]M. Antikainen, T. Aura and M. Särelä, "Denial-of-Service Attacks in Bloom-Filter-Based Forwarding," in IEEE/ACM Transactions on Networking, vol. 22, no. 5, pp. 1463-1476, Oct. 2014, doi: 10.1109/TNET.2013.2281614

[6] P. Reviriego, S. Pontarelli, J. A. Maestro and M. Ottavi, "A Synergetic Use of Bloom Filters for Error Detection and Correction," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 3, pp. 584-587, March 2015, doi: 10.1109/TVLSI.2014.2311234

[7] X. Liu et al., "Sampling Bloom Filter-Based Detection of Unknown RFID Tags," in IEEE Transactions on Communications, vol. 63, no. 4, pp. 1432-1442, April 2015, doi: 10.1109/TCOMM.2015.2402660.

[8] K. Lin, H. Chen, T. Dai, D. Liu, L. Liu and L. Shi, "Segmented Bloom Filter Based Missing Tag Detection for Large-Scale RFID Systems With Unknown Tags," in IEEE Access, vol. 6, pp. 54435-54446, 2018, doi: 10.1109/ACCESS.2018.2872543

[9] H. Byun, S. Kim, C. Yim and H. Lim, "Addition of a Secondary Functional Bloom Filter," in IEEE Communications Letters, vol. 24, no. 10, pp. 2123-2127, Oct. 2020, doi: 10.1109/LCOMM.2020.3003695.

[10] R. -H. Shi, "Quantum Bloom Filter and Its Applications," in IEEE Transactions on Quantum Engineering, vol. 2, pp. 1-11, 2021, Art no. 2100411, doi: 10.1109/TQE.2021.3054623.

[11] D. Zheng, A. Wu, Y. Zhang and Q. Zhao, "Efficient and Privacy-Preserving Medical Data Sharing in Internet of Things With Limited Computing Power," in IEEE Access, vol. 6, pp. 28019-28027, 2018, doi: 10.1109/ACCESS.2018.2840504.

[12] J. Jeong, J. W. J. Joo, Y. Lee and Y. Son, "Secure Cloud Storage Service Using Bloom Filters for the Internet of Things," in IEEE Access, vol. 7, pp. 60897-60907, 2019, doi: 10.1109/ACCESS.2019.2915576.