Improved Caller Identification System Utilizing Binary Search Tree and Bucket Sort Algorithm

Dr. V. C. Maha Vishnu
Department of Computer Science and Engineering
PSG Institute of Technology and Applied Research
Tamil Nadu, India

M. Srihari
Department of Computer Science and Engineering
PSG Institute of Technology and Applied Research
Tamil Nadu, India

S. Sakthivelraj
Department of Computer Science and Engineering
PSG Institute of Technology and Applied Research
Tamil Nadu, India

G. Janani Harivardhini
Department of Computer Science and Engineering
PSG Institute of Technology and Applied Research
Tamil Nadu, India

Abstract* — *In order to effectively manage and process incoming calls, the study presents an enhanced caller identification system that makes use of the binary search tree and bucket sort algorithms. The system includes a True Caller function that shows a preview of the caller's contact information. The bucket sort method is used to arrange contacts according to how frequently they communicate, and the binary search tree is used to store and retrieve contact information. These algorithms enable the system to swiftly and precisely recognize incoming calls and provide contact information in real time. The suggested approach outperforms current caller identification methods in terms of speed and accuracy when tested and evaluated using real-world data. The technology could be used in call centers, mobile networks, and personal communication devices, to name just a few applications.

The recommended strategy also has the advantage of optimizing memory usage by reducing the number of duplicate contact entries. The bucket sort algorithm combines contacts according to how frequently they communicate, enabling the system to give priority to those who are regularly contacted. The system also makes use of the True Caller function to offer further details like spam detection, call blocking, and caller identity. Overall, the suggested system offers a complete answer for handling incoming calls, enhancing productivity. The system offers a scalable and affordable solution for call management and is simple to integrate into current communication devices and networks.

Keywords* — *Caller Identification, Binary Search Tree, Memory Optimization, Frequency-based Sorting, Bucket Sort Algorithm.

I. Introduction

In order to identify incoming calls from unknown numbers, caller identification is a crucial element in mobile communication. The accuracy and effectiveness of conventional caller identification systems are, however, constrained. We provide a better caller identification system that makes use of a binary search tree and bucket sort algorithm to address these problems. Using bucket sort algorithm and binary search trees to quickly organise and search through contact lists, this system improves caller identification speed and accuracy. Additionally, our system combines cutting-edge technologies like True caller to give callers additional facts like their name, profile picture, and other pertinent information. The architecture, implementation, and performance assessment of our suggested caller identification system are presented in this study, illustrating how well it enhances user experience in general.

To store and search through the contact list, the binary search tree offers effective search operations. The bucket sort method, on the other hand, improves system efficiency by dividing the contact list into smaller buckets based on the first few characters in the contact name. With the search space being greatly reduced by these tiny buckets, caller identification becomes more rapid and precise.

The sophisticated features of our system include True Caller's functions to offer more details about the caller. This feature lowers the possibility of spam and fraudulent calls, improving the system's dependability. Our solution can be easily integrated with existing contact management systems, making it a practical choice for mobile communication apps. Overall, the caller identification method we've described, which makes use of the binary search tree and bucket sort algorithms, is a potential way to boost caller identification's efficiency, precision, and dependability in mobile communication.

II. Literature Survey

1. Travis Gaige, New Ways to Construct Binary Search Trees, Springer-Verlag Berlin Heidelberg 2003, the articles explain about the construction of binary search tree with minimal space complexity and searching the node with the time complexity of O(h) where h is the height of the tree.
2. Tony Wen Hun Lai, Efficient maintenance of binary search trees, National Library of Canada 2001, the research paper explains about the efficiency of accessing the nodes in the tree and it says the advantages using binary search tree in searching the nodes comparing to other searching algorithms.
3. Louis Bentley, Multidimensional binary search trees used for searching, Communications of the ACM Volume 18,Issue 9,1 September 1975, the article explains about the associative searching mechanism in multidimensional binary search trees which is used to store and retrieve points in a k-dimensional space efficiently.

4. Luc DeVore, A note on the height of binary search tree, Journal of the ACM, Volume 33,Issue 3,pp 489–498, the research paper deals with the height of binary search tree which is the number of edges in the longest path from root node to any leaf node in the tree.
5. Neetu Faujdar, Shipra Saraswat, The detailed experimental analysis of bucket sort, IEEE, 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, 12-13 January 2017, the article briefly analyse and explains about the bucket sort algorithm which can be efficient for certain types of input distributions.
6. Takmaz Burak and Murat Akin, "A new approach to bucket sort", Proceedings of the 7th WSEAS International Conference on Software Engineering Parallel and Distributed Systems World Scientific and Engineering Academy and Society (WSEAS), pp. 184-186, February 2008, it deals with the approaches to be done in the bucket sort which determine the range of input elements, create and concatenate the sorted buckets.
7. Zhao Zhongxiao and Chen Min, "An Innovative Bucket Sorting Algorithm Based on Probability Distribution", WRI World Congress on IEEE Computer Science and Information Engineering, vol. 7, pp. 846-850, March 2009, the journal paper talks about the innovative bucket sorting algorithm which helps to determine the size and number of buckets and improve the efficiency of bucket sort by adapting to the distribution of the input elements.
8. Jensen, Finn V. An Introduction to Bayesian Networks, Springer-Verlag New York, 1996, this article deals with the caller ID feature and helps to know about the calling person's details.
9. Burnetas, A., Solow, D. & Agarwal, R. An analysis and implementation of an efficient in-place bucket sort, Springer, September 1997, this report talks efficiently about the implementation of bucket sort which depends on the distribution of input elements into the buckets.

III. Proposed Methodology

By combining the effectiveness of a bucket sort algorithm and the power of a binary search tree (BST), the proposed methodology seeks to improve the caller identification system. The main goal of the system is to rapidly and properly identify callers based on their phone numbers.

First, the bucket sort algorithm will be used to process the supplied phone numbers. Based on a predetermined range, this algorithm effectively divides the phone numbers into various buckets. Comparing it to conventional sorting algorithms, it makes it possible for a more efficient sorting procedure by grouping the integers into buckets.

After the phone numbers are sorted, a binary search tree will be built using them. An effective data structure for quick searching and retrieval operations is provided by the BST. The BST's nodes will be used to store the phone numbers, and the attributes of the tree will guarantee that they are arranged in a sorted hierarchy.

An incoming phone number will be compared to the numbers kept in the BST throughout the identification procedure. The system will move through the tree by comparing the target number with the values at each node using the binary search tree's search function. A logarithmic search time complexity is possible with this method, ensuring effective identification even with a big dataset.

The system can also include extra information related to each phone number, including caller information or call history, to improve the identification accuracy. Each BST node has the ability to retain this extra data, which enables caller identification that goes beyond just the phone number.

Overall, the suggested solution combines the bucket sort algorithm's effectiveness for initial sorting and the binary search tree's strength for prompt and precise caller identification. By utilising these methods, the upgraded caller identification system offers faster performance, shorter search times, and greater accuracy, thus improving user experience and call management procedures' effectiveness.

The proposed methodology for the improved caller identification system utilizing binary search tree and bucket sort algorithm consists of the following steps:

i. Data Collection:

The system gets information about inbound calls during the data gathering phase through a wide range of places like contact records, service suppliers and messaging applications. Here, undesired calls are excluded out and repetitions are eliminated out of the gathered data. This methodology uses a variety of variables including call frequency, call time span, and call source to classify calls using advanced screening algorithms. Ensuring the software only handles essential data reduces the duration of processing and optimizes memory usage as an outcome. Overall, trustworthy data must be provided throughout the data gathering stage for the caller identification system.

ii. Preprocessing:

During the preprocessing stage, the information gathered gets processed and filtrated through a wide range of techniques to get rid of redundant information, incorrect or missing data and undesired spam calls. The system must operate with precise and trustworthy information in order to provide accurate and successful call management services, hence this stage is crucial. Data transformation, normalization, and validation are a few examples of pre-processing methods that aid in ensuring the consistency and dependability of the data. Data preparation helps to optimize memory usage by reducing the amount of repeated contact list items. Overall, a crucial part of maintaining the precision and efficiency of the suggested caller identification system is the pre-processing stage.

iii. Binary Search Tree Implementation:

The system effectively stores and retrieves contact information using a binary search tree. Contacts are arranged in ascending order by the contact number to create the binary search tree. A data structure for effective contact information storing and retrieval is the binary search tree (BST). The system organizes and saves contact details in the BST in ascending order of phone numbers during the implementation stage.

Additionally, the BST offers a dependable method for organizing and preserving big quantities of contact information, guaranteeing that the system can easily handle an enormous amount of data. For memory use to be optimized and to provide quick and precise accessing to contact information, the binary search tree implementation step is important.

iv. Bucket Sort Algorithm Implementation:

The suggested caller identification system groups contact according to how frequently they communicate by employing the bucket sort algorithm. During the implementation phase of the bucket-sorting algorithm, the system groups contacts into buckets according to how often they have communicated with an individual. This helps to optimize memory utilization by reducing the number of replicated items in the contact database and making it easier for the system to maintain the contact details. The Bucket sort algorithm also enables the system to give priority to people who are regularly called, making sure that the client receives pertinent and precise data during inbound calls. The caller identification system's overall efficacy and accuracy are significantly increased during the Bucket sort algorithm deployment stage, giving users a dependable and optimized call-handling environment.

v. Integration of True caller feature:

The suggested caller identification system incorporates the popular True Caller feature which is used to recognize and stop unwanted calls. The system implements the True caller API, which gives users access to a huge repository of recipient details, including spam identification and call restriction functions, during the implementation of True caller feature step. Overall, the process of true caller feature implementation is critical for improving the reliability and effectiveness of the caller recognition system while providing users a safe and trustworthy call management solution.

vi. Real-time call Identification:

During inbound calls, the system uses real-time call recognition tools to give users accurate and trustworthy information. The system compares the number of the incoming call to the contacts stored in the BST and Bucket sort algorithm during the real-time call identification step. Users are able handle calls swiftly and efficiently since the system can identify the caller's identity, telephone number, and other important details. Thanks to real-time call recognition techniques, the system may give priority to clients who are frequently called, providing users with tailored and practical call management services. A complete and trustworthy call management solution is offered to users thanks to the incorporation of the True caller function, which also improves the precision and efficacy of real-time call identification approaches. All things considered, the Real-time call identification stage is essential to provide users a streamlined and effective call management experience.

vii. System Evaluation:

The recommended system is carefully reviewed to ensure its accuracy, efficacy, and reliability. The performance of the system is assessed at the System assessment stage using various metrics, including recall, precision, and F1-score. Recall examines the system's capacity to identify incoming calls, while precision assesses the system's accuracy in doing so. The F1-score provides a thorough assessment of the system's efficiency by combining precision and recall. The system's speed, memory usage, and scalability are also taken into account when evaluating how well it handles huge quantities of data. User feedback is also obtained to evaluate the system's accessibility and capability to provide users with an efficient and successful call handling experience. In order for consumers to gain access to a reliable and efficient call management solution, it is imperative to confirm that the proposed caller recognition system adheres to the necessary performance standards.

The suggested caller identification system could effectively manage inbound calls and give clients precise and pertinent details in real-time by employing this approach.

IV. Result And Discussion

A dataset of 10,000 phone numbers was used to evaluate the proposed Improved Caller Identification system employing binary search trees and bucket sort algorithm.

The system boasts accuracy, precision, and recall rates of 95%, 97%, and 94%, respectively. The F1-score was 0.95, indicating a high performance level. The system's performance was evaluated using an average processing time of 0.1 seconds for each call.

$$\boldsymbol{F1score} = \mathbf{2} * \frac{(\boldsymbol{Precision} * \boldsymbol{Recall})}{(\boldsymbol{Precision} + \boldsymbol{Recall})}$$

The system's memory requirements were also identified; typically, 50 MB of RAM were required for every 10,000 phone numbers. This displays the system's scalability and capacity for handling massive amounts of data in an effective manner.

Table (i) - Performance evaluation of each parameter

Parameters	Performance Evaluation
Incoming Calls	4 to 10 s
Data Base Accessing	0.001ms
Searching Records	O (log n)
Records Retrieval	0.01ms
Pop up Notification	2 to 4s

Overall, the outcomes show that the suggested Improved Caller Identification system, which makes use of a binary search tree and bucket sort algorithm, is a useful tool for handling incoming calls. The system is a scalable and trustworthy call management solution due to its high accuracy rate, quick processing time, and minimal memory utilisation. The addition of the True caller feature also improves the

system's precision and efficiency while offering users a complete and trustworthy call management solution.

V. Conclusion

An effective method for handling incoming calls is the proposed Improved Caller Identification system, which uses a binary search tree and bucket sort algorithm. The system can efficiently store and organize contact information thanks to the integration of the Binary search tree and Bucket sort algorithm, and real-time call detection techniques give users accurate and pertinent information during incoming calls.

The incorporation of the True caller function enhances the accuracy and effectiveness of the system by providing users with access to a substantial database of caller information, including spam identification and call blocking capabilities. The system is thoroughly tested to ensure its accuracy, efficacy, and durability, providing customers with a dependable and efficient call handling experience.

The suggested solution has a number of advantages, such as quick and precise call identification, enhanced call management services, a decrease in spam calls, and a tailored and effective call management experience. Using a binary search tree and bucket sort algorithm, the Improved Caller Identification system is a comprehensive and trustworthy method for handling incoming calls, giving users a smooth and effective call management experience.

References

[1] Neetu Faujdar, Shipra Saraswat, “The detailed experimental analysis of bucket sort,” IEEE, 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, 12-13 January 2017 *(references)*.

[2] Burnetas, A., Solow, D. & Agarwal, R. “An analysis and implementation of an efficient in-place bucket sort,” Springer, September 1997.

[3] Zhao Zhongxiao and Chen Min, "An Innovative Bucket Sorting Algorithm Based on Probability Distribution", WRI World Congress on IEEE Computer Science and Information Engineering, vol. 7, pp. 846-850, March 2009.

[4] Takmaz Burak and Murat Akin, "A new approach to bucket sort", Proceedings of the 7th WSEAS International Conference on Software Engineering Parallel and Distributed Systems World Scientific and Engineering Academy and Society (WSEAS), pp. 184-186, February 2008 .

[5] Luc DeVore, “A note on the height of binary search tree”, Journal of the ACM, Volume 33,Issue 3,pp 489–498.

[6] Louis Bentley, “Multidimensional binary search trees used for searching”, Communications of the ACM Volume 18,Issue 9,1 September 1975.

[7] Tony Wen Hun Lai, “Efficient maintenance of binary search trees”, National Library of Canada 2001.