International Journal of Computing Sciences Research (ISSN print: 2546-0552; ISSN online: 2546-115X)
Vol. 8, pp. 3088-3102
doi: 10.25147/ijcsr.2017.001.1.205
https://stepacademic.net

Short Paper*

Enhancement of Rabin-Karp Algorithm using XOR Filter

Beatrice A. Escoto
BS Computer Science, Pamantasan ng Lungsod ng Maynila, Philippines
baescoto2020@plm.edu.ph
(corresponding author)

Acemon Jassene G. Ladines
BS Computer Science, Pamantasan ng Lungsod ng Maynila, Philippines
ajgladines2020@plm.edu.ph
(corresponding author)

Abigail Grace G. Macahilig
BS Computer Science, Pamantasan ng Lungsod ng Maynila, Philippines
aggmacahilig2020@plm.edu.ph
(corresponding author)

Mark Anthony S. Mercado
Information Technology Department, Pamantasan ng Lungsod ng Maynila, Philippines
masmercado@plm.edu.ph

Dan Michael A. Cortez
Computer Science Department, Pamantasan ng Lungsod ng Maynila, Philippines
dmacortez@plm.edu.ph

Raymund M. Dioses
Computer Science Department, Pamantasan ng Lungsod ng Maynila, Philippines
rmdioses@plm.edu.ph

Jonathan C. Morano
Computer Science Department, Pamantasan ng Lungsod ng Maynila, Philippines
jcmorano@plm.edu.ph

Jamillah S. Guialil
Computer Science Department, Pamantasan ng Lungsod ng Maynila, Philippines
jsguialil@plm.edu.ph

Date received: May 7, 2024
Date received in revised form: July 5, 2024
Date accepted: July 5, 2024

Recommended citation:

Escoto, B., Ladines, A. J., Macahilig, A. G., Mercado, M. A., Cortez, D. M., Dioses, R., Morano, J., & Guialil, J. (2024). Enhancement of Rabin-Karp Algorithm using XOR filter. *International Journal of Computing Sciences Research*, *8*, 3088-3102. https://doi.org/10.25147/ijcsr.2017.001.1.205

**Special Issue on International Research Conference on Computer Engineering and Technology Education (IRCCETE). Guest Associate Editors: Dr. Roben A. Juanatas (National University-Manila) and Dr. Nelson C. Rodelas (University of East).*

Abstract

Purpose – The study aims to enhance the Rabin-Karp Algorithm that underlines the problem encountered wherein the algorithm's runtime performance is affected due to the continuous rapid growth of data.

Method – Application of XOR Filter in the enhancement of the Rabin-Karp Algorithm is constructed to the given patterns to check for any pattern absences that will be removed from the data input. Then the updated patterns will be utilized in the string-matching process.

Results – The modified method displayed significant improvements in different input data sizes and patterns. After conducting runtime tests, it surpasses the improved algorithm using Bloom Filter by 9%, and a rate of 47.53% runtime performance compared to the traditional Rabin-Karp Algorithm.

Conclusion – The integration of the XOR Filter into the Rabin-Karp Algorithm, demonstrates a statistically significant runtime improvement. Moreover, it shows an effective scalability with larger datasets, proving its practical suitability for applications or scenarios that handle extensive datasets.

Recommendations – Future researchers are encouraged to explore other alternative techniques and the use of different filters that are not mentioned in this study. In addition, it extends future research on the use of Artificial Intelligence to explore diverse strategies and implementation for further improvement of the modified algorithm.

Research Implications – Potential directions for further research in string-matching algorithm optimization on effective integration of the XOR Filter into the Rabin-Karp Algorithm. Further highlighting the possibilities of using probabilistic data structures in algorithmic design are observed with the improvements in runtime efficiency. Data processing and pattern-matching areas open the door to effective and scalable solutions by promoting studies in the use of cutting-edge AI techniques to improve current algorithms.

Practical Implications – The Enhanced Rabin-Karp Algorithm can be used by Academic Institutions and Future Researchers to provide a more accurate string-matching process and improved runtime, supporting applications working with massive datasets.

Keywords – string matching, probabilistic data structure, bloom filter, pattern, Rabin-Karp, XOR filter

INTRODUCTION

String matching is one of the fundamental concepts in computer science, that is widely utilized in applications like text processing, pattern matching, and information retrieval. Its significance extends to multimedia databases (Hakak et al, 2019), where it facilitates the extraction of contextually significant words for categorization and indexing. This process, enables deeper queries, optimizing information retrieval (Lubis, Ikhwan, and Kan, 2018) (Janani and Vijayarani, 2016).

It plays a significant role in theoretical science (Faro and Lecroq, 2013), presenting challenging problems that entail the application of fundamental concepts to practical software implementation. String matching can be categorized into two types: a single pattern match, which involves a single parameter; and a multi pattern match, where numerous parameters are considered (Buchanan, n.d). Various established algorithms exist for finding these patterns (single and/or multiple patterns), including the Brute Force Method (Hakak et al, 2018), Knuth-Morris-Pratt (KMP) (Rahim, Zulkarnain, and Jaya, 2017), Boyer-Moore (Rahim et al, 2017), Deterministic Finite Automaton Method (DFA) (BabuKaruppiah and Rajaram, 2011), and the Rabin-Karp Algorithm (Siahaan, 2018).

As for the context of this study, one of the widely recognized and utilized techniques for string matching is the Rabin-Karp algorithm used in various applications. The RKA employs a distinctive approach where it transforms strings to hash values to distinctly identify similar patterns.

The study that was conducted by Moeini and Shahhoseini (2022), gave the researchers the interest to enhance the improved Rabin-Karp algorithm. Their study was motivated by their previous study (Moeini and Shahhoseini, 2019) wherein they prevented the redundancy of comparisons and accelerated the speed performance in the matching process. In their 2022 study, they incorporated a type of probabilistic data structure which is the Bloom Filter to the Rabin-Karp. This is where they stated the problem of most traditional algorithms are not sufficient in holding large data volumes. The results show the comparison of the enhanced method to verify its impact on the improvement of the Rabin-Karp algorithm.

With the insertion of the XOR Filter due to its one of the fastest and less memory consumption than other filters (Martinez and Larrabeiti, 2024), it will be placed in the preprocessing phase of the Rabin-Karp. This will give probable results of the researcher's method a better performance for the runtime execution compared to previous techniques made in the improvement of the Rabin-Karp Algorithm. A study by Graf and Lemire (2020) evaluates the performance of three filters: Bloom, Cuckoo, and XOR Filters. They mentioned the problems encountered in both Bloom and Cuckoo Filters, while XOR Filter has not yet been implemented in any previous studies. From their evaluation, they concluded that the use of XOR Filters is faster and smaller than the two filters mentioned.

Therefore, the objective of this study is to enhance the Rabin-Karp Algorithm by incorporating the XOR Filter. This aims to the optimization of the algorithm's runtime performance and has a detailed comparative analysis table showcasing the outcomes and efficiencies of the integrated filters to the said algorithm.

LITERATURE REVIEW

Rabin-Karp Algorithm

The Rabin-Karp algorithm created by Michael O. Rabin and Richard M. Karp in 1987 is a type of string-matching algorithm that searches through a text for a substring pattern using hashing technique and brute force comparison. This algorithm works exceptionally well in multiple-string matching (Putri and Siahaan, 2017).

From the study of Leman, Rahman, Ikorasaki, Riza, & Akbbar (2019) it was shown how the Rabin-Karp works as follows:

1. Application of hash functions
2. Preprocessing phase in time complexity O(m) and constant place
3. Seeking phase in the complexity O(m)
4. O(n+m) estimation of active time second paragraph

It is discussed that indexing and retrieving data are based on keys by the hash function. The hash function assigns the index value to a key, and every time the data relates to the key, it has its access, and the hash function is applied. First, the text is hashed when the data is saved, which may result in a value. The value is then compared to the index value when it seeks the data.

Probabilistic Data Structure

Probabilistic Data Structures or PDS are essential for optimizing Big Data and streaming applications, reducing analytical latency. These different data structures use hash functions to efficiently represent sets of objects in stream-based computing, providing approximations with error boundaries. This guarantees that precise approximations are incorporated directly into the collection of data. PDS outperforms traditional data structures by utilizing less memory while maintaining constant processing time for complex queries (Singh et al, 2020). This collection of different data structures includes the Bloom filter (Luo et al, 2020), the Cuckoo filter (Fan et al, 2014), and the XOR filter (Graf and Lemire, 2020) which is widely used in different applications.

Bloom Filter

Introduced by Burton Howard Bloom in 1970, A Bloom filter is a space-efficient probabilistic data structure that checks whether a given element belongs to a set or not. Its primary advantage is its ability to deliver constant-time membership queries with a relatively small amount of memory. Bloom filters are widely employed in applications where the cost of false positives is acceptable and memory efficiency is critical, such as network routers, distributed systems, and databases (Luo et al, 2020).

Here is a quick rundown of how a Bloom filter functions:

1. Several hash functions and a fixed-size array (bit array) of zeroes are used to initialize a Bloom Filter
2. Then it hashes every element introduced to the set using a variety of hash functions. The relevant bits in the array the set to 1 using the hash values that are generated by the filter.
3. The same hash functions are also to determine whether an element is a part of the set. The element is most likely to be included in the collision it could lead to producing false positives.
4. Due to the instance of a hash collision which could give false positives, implying that an element is in the set when it is not. Bloom filter, however, never produces false negatives; if it indicates that an element is not in the set, it is accurate.

METHODOLOGY

The Rabin-Karp algorithm will be using the XOR Filter in this study to improve its capabilities and solve the execution time problem in the algorithm.

In 2020, Guaf and Lumire conducted an initial study on the XOR Filter to give a space-efficient substitute for traditional data structures such as Bloom filters, while maintaining low false positive rates and constant-time lookups. The XOR filter, which combines bitwise XOR operation with hash functions, is described in the study as a probabilistic data structure that is utilized for approximate set membership checking. A set of items can be efficiently represented by it, and it can ascertain the likelihood of any given element being a member of the set or not.

XOR Filter will help resolve the runtime issues encountered by the Rabin-Karp Algorithm because of its constant-time complexity for set membership testing, which implies that the time spent to check whether an element is a member of the set does not rely on the size of the dataset. When the XOR Filter was put to the test against other probabilistic data structures, it performed very well in terms of query-time efficiency and having little space consumption, even for moderate false-positive probability.

To enhance the runtime efficiency, there are several optimization techniques for the traditional Rabin-Karp algorithm. As for this method, employing the XOR filter before applying the Rabin-Karp is introduced to improve the algorithm's execution time. The flowchart of the enhancement on the Rabin-Karp Algorithm can be seen in Figure 1.

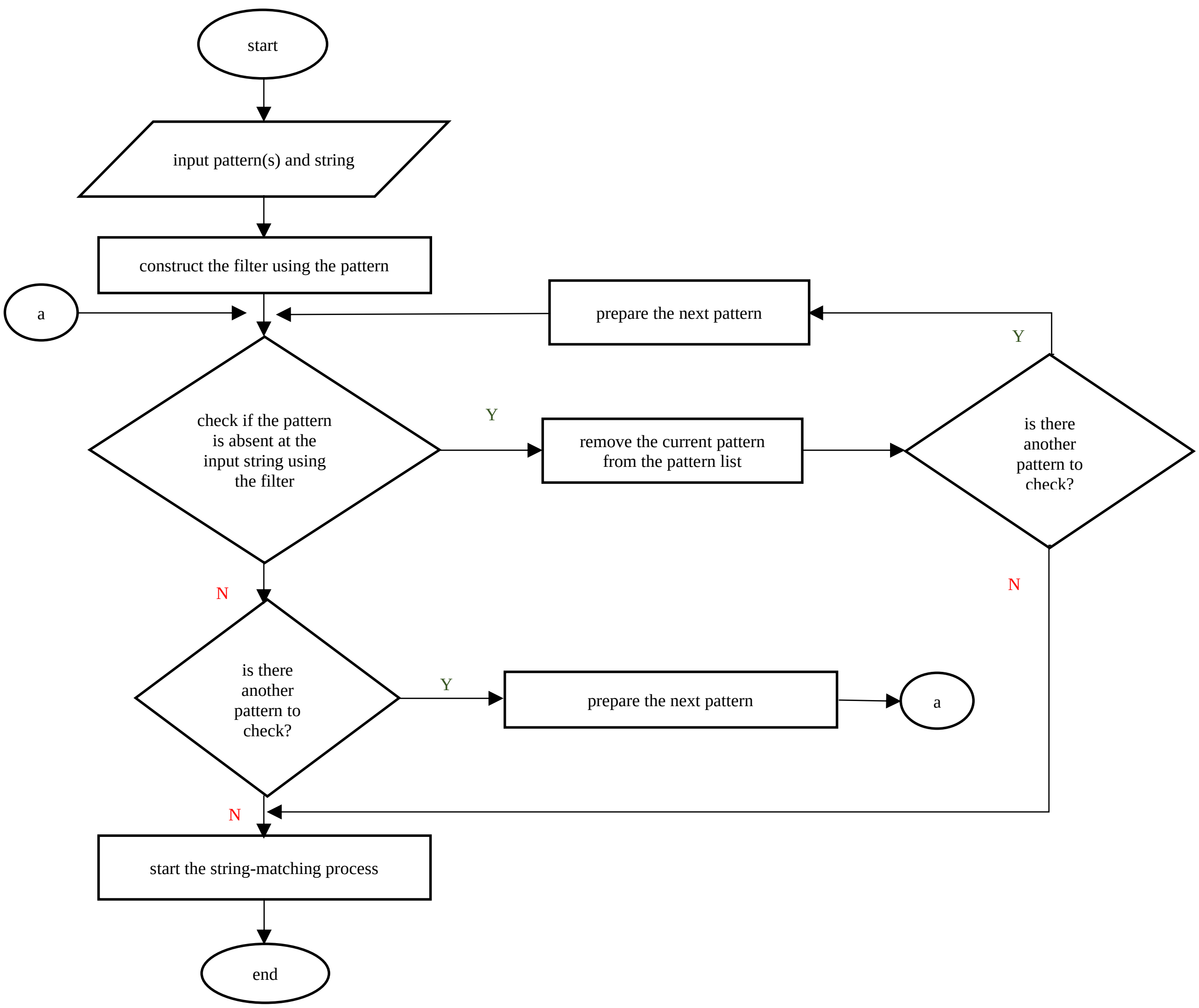

Figure 1. Flowchart of the Enhanced Method

The enhanced method has been implemented through the following steps:

- The XOR Filter is constructed using the given patterns, which will be used to check for any pattern absence in the given input data.
- All the patterns that are not present in the input data will be removed, resulting in a decrease in patterns subject to comparison.
- The string-matching process will now be performed using the updated patterns, sliding through the input text until it reaches the final character.

Rabin-Karp now allows for faster completion of the exact matching step. Comparing all patterns is not necessary, resulting in fewer comparisons and a lower computing load for multi-pattern string matching. Additionally, it boosts performance. Furthermore, the frequency of memory calls lowers significantly.

The Enhanced Rabin-Karp Algorithm using XOR Filter was evaluated using randomly generated strings for the pattern and input data. The length varied from 81920 bits up to 5242880 bits, in contrast, the size used was chosen to assess the enhanced algorithm performance across different data sizes. Lastly, for the tool used to test the runtime of the algorithm, we used the 'time' library in Python, in which a function called process time is available and capable of computing the CPU time of a program.

RESULTS

The enhanced algorithm was written in Python programming language and was executed in a Linux environment (Ubuntu) using Windows Subsystem for Linux (WSL) on a Ryzen 3 CPU. Furthermore, comparing the existing target algorithm to the enhanced algorithm was evaluated using a runtime test, this refers to the time taken by the algorithm to execute. The chosen method conducted to measure the runtime of the algorithm does not include the time taken during sleep, as it only renders the time that the current process rendered since it was generated by the operating system. This way of calculating the runtime of an algorithm acts as a portable counter but also offers the best available resolution.

The input parameters used in the algorithm as shown in Table 1 are represented in terms of bit(s) and consist of the following:

- *Input Data* – The string that will be scanned or searched through for any pattern that matches the given ‘pattern’ parameter.
- *Pattern Data* – The string or pattern that will be used as the basis for searching and matching the strings within the Input Data.

Table 1. List of Execution Time for the Naïve RKA, Target Algorithm, and the Enhanced Algorithm

Input Data (B)	***EXECUTION TIME(S)*** *(lower is better)*		
	Rabin Karp Algorithm	**Improved RKA Using Bloom Filter**	**Modified Rabin-Karp Algorithm**
81920	11	7	6
163840	46	26	24
372680	182	101	94
655360	712	400	376
1310720	2775	1595	1447
2621440	11403	6275	5854
5242880	43920	26297	23120

Table 1 displays a list of execution time (S) of the Naïve RKA, Improved RKA using Bloom Filter, and the Modified Rabin-Karp Algorithm. The leftmost column shows different sizes of input data (B). Having a lower record of time, this gives better results which will be also presented in the following figures.

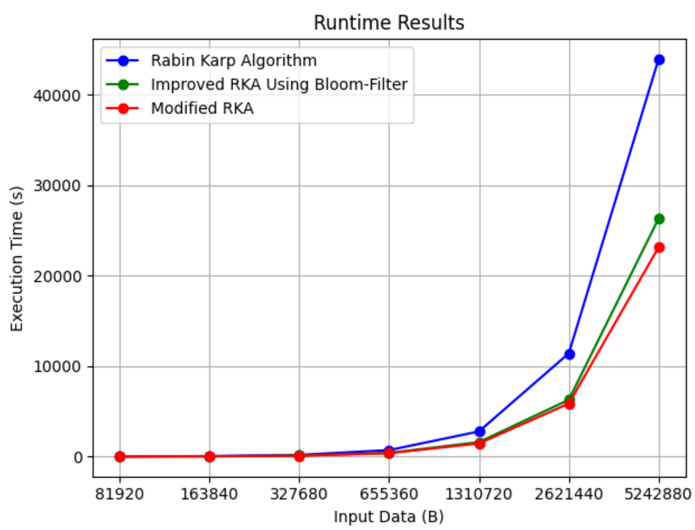

Figure 2. Runtime Results per Input Data

Figure 2 which is the line graph that represents Table 1, compares the runtime results for various input data sizes for the Naïve Rabin Karp Algorithm, the Improved Rabin Karp Algorithm with the Bloom Filter, and the enhanced algorithm speed in terms of execution time.

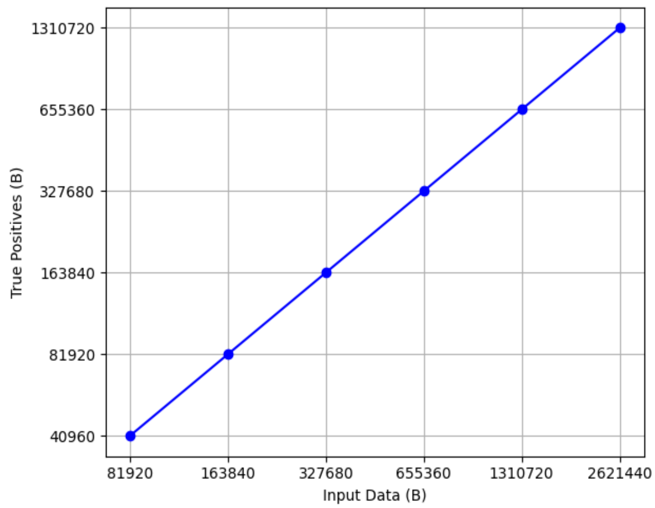

Figure 3. Number of true positive patterns per input data

Regarding Table 1, the researchers generated uniform testing parameters, all tests provided contain the same data and pattern. It is also important to note that the size of the given pattern is equal to the size of the input string, in addition to that it also consists of two subsets, one having all true positives (present in the input string) as shown in Figure 3, and the other does not.

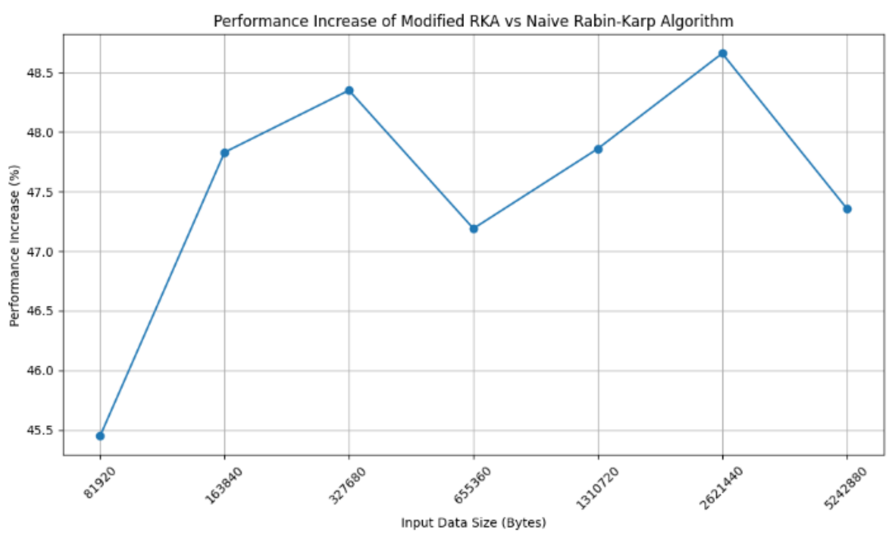

Figure 4. Percentage of the increased performance of the enhanced algorithm compared to the naïve Rabin-Karp algorithm.

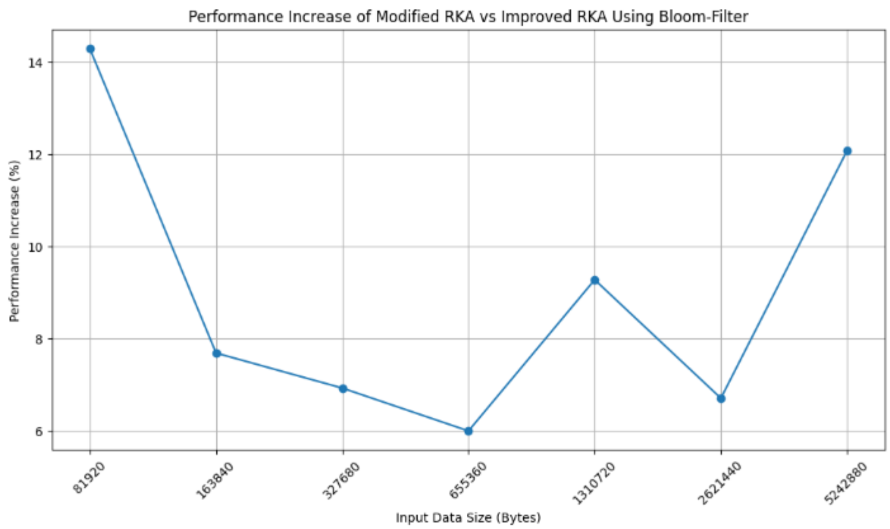

Figure 5. Percentage of the increased performance of the enhanced algorithm compared to the Improved Rabin-Karp Using Bloom-Filter.

DISCUSSION

Upon closer analysis of the results presented in Table 1, and Figures 4 and 5, the efficiency of the enhanced algorithm consistently outperforms the Naïve and Improved RKA using Bloom Filter. This implies that the enhanced enhancement maintains its performance well in handling different datasets, prevailing it as practically suitable for real-world applications where processing efficiency is crucial. Moreover, the technique performs better in every tested scenario; yet the effectiveness of the enhancement varies from what is introduced depending on the specific characteristics of the input data and pattern length. This underlines the importance of considering the nature of data, and the requirements of an application when selecting an algorithm for pattern or string-matching tasks.

Lastly, it shows a significant improvement in the runtime performance of the enhanced algorithm compared to the existing methodologies. On average results, the enhanced algorithm surpasses the Improved Rabin-Karp using Bloom Filter by 9%, and notably, it achieves a substantial 47.53% runtime performance gained compared to the baseline of the Rabin-Karp algorithm.

CONCLUSIONS AND RECOMMENDATIONS

In conclusion, the integration of the XOR Filter into the Rabin-Karp Algorithm shows runtime efficiency improvements compared to the targeted approach which uses the Bloom Filter. The modified solution shows its ability to scale effectively with larger data sets, making it practically suitable for applications or scenarios that deal with massive data inputs. Leveraging probabilistic data structures like the XOR Filter optimizes pattern-matching algorithms, reducing unnecessary computations by actively checking the presence and/or absence of the given pattern in the input data. However, performance gains vary based on data characteristics like input size and pattern length, emphasizing the importance of careful algorithm selection and implementation.

The researchers strongly recommend future researchers delve into alternative techniques that were not addressed within the scope of this study. In addition, other types of filters may be utilized depending on the implementation that will be done for future works. Lastly, there is a notable surge in the development of Artificial Intelligence (AI) where future researchers can explore diverse strategies and implementations aiming for the improvement of the current algorithm. Overall, this study potentially offers promising implications for efficient pattern matching in various domains, warranting further optimization and exploration in future research endeavors.

IMPLICATIONS

The finding of this study left its significance for both research and practical applications in the field of pattern-matching algorithms. With an observed improvement in runtime efficiency by incorporating probabilistic structures, such as XOR Filter, it demonstrates potential subtle modifications in producing authentic advancements, especially in dealing with massive amounts of data. However, variations in performance increase across data features emphasizing the relevance of carefully selecting and implementing algorithms adapted to certain scenarios. Furthermore, the proposal for exploring alternative methodologies and filters that are not included in this work opens possibilities for future research, potentially leading to major advances in pattern-matching algorithms. Collectively, this study lays the framework for future research aimed at improving efficiency and effectiveness in a variety of disciplines.

ACKNOWLEDGEMENT

The authors express sincere appreciation to Professor Mark Anthony S. Mercado for his guidance and expertise, as well as to Doc Dan Michael A. Cortez, Prof. Raymund M. Dioses, Prof. Jonathan C. Morano, and Prof. Jamillah S. Gualil for their invaluable assistance and feedback. Their combined efforts played a pivotal role in molding the study, and their contributions are much valued. The authors would also like to thank Pamantasan ng Lungsod ng Maynila's College of Information System and Technology Management and the Computer Science Department for their cooperation in offering great knowledge and valuable insight to the study.

FUNDING

The study did not receive funding from any institution.

DECLARATIONS

Conflict of Interest

The researcher declares no conflict of interest in this study.

Informed Consent

Informed consent does not apply to this study because it involves the analysis of publicly available data that does not include identifiable personal information.

Ethics Approval

Ethics approval is not applicable in this study as it involves public publicly accessible information analysis. Any data collected will be treated under the relevant regulations and best practices from the fields that are covered by this research.

REFERENCES

BabuKaruppiah, A., & Rajaram, S. (2011, March). Deterministic finite automata for pattern matching in FPGA for intrusion detection. In *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)* (pp. 167-170). IEEE.

Buchanan, B. (n.d). *CSE 5717 Final Paper I. Problem Statement/Abstract: One of the classic problems in Computer Science is string matching.* Retrieved from http://benjibuchanan.com/img/buchananFinalPaper.pdf

Fan, B., Andersen, D. G., Kaminsky, M., & Mitzenmacher, M. D. (2014, December). Cuckoo filter: Practically better than bloom. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies* (pp. 75-88). ACM.

Faro, S., & Lecroq, T. (2013). The exact online string matching problem: A review of the most recent results. *ACM Computing Surveys (CSUR)*, *45*(2), 1-42.

Graf, T. M., & Lemire, D. (2020). Xor filters: Faster and smaller than bloom and cuckoo filters. *Journal of Experimental Algorithmics (JEA)*, *25*, 1-16.

Hakak, S., Kamsin, A., Shivakumara, P., Idna Idris, M. Y., & Gilkar, G. A. (2018). A new split-based searching for exact pattern matching for natural texts. *PloS one*, *13*(7), e0200912.

Janani, R., & Vijayarani, S. (2016). An efficient text pattern matching algorithm for retrieving information from desktop. *Indian Journal of Science and Technology*, *9*(43), 1-11.

Leman, D., Rahman, M., Ikorasaki, F., Riza, B. S., & Akbbar, M. B. (2019, November). Rabin Karp and Winnowing algorithm for Statistics of text document plagiarism detection. In *2019 7th International Conference on Cyber and IT Service Management (CITSM)* (Vol. 7, pp. 1-5). IEEE.

Lubis, A. H., Ikhwan, A., & Kan, P. L. E. (2018). Combination of Levenshtein distance and rabin-karp to improve the accuracy of document equivalence level. *International Journal of Engineering & Technology*, *7*(2.27), 17-21.

Luo, L., Guo, D., Ma, R. T., Rottenstreich, O., & Luo, X. (2018). Optimizing bloom filter: Challenges, solutions, and comparisons. *IEEE Communications Surveys & Tutorials, 21*(2), 1912-1949.

Martínez, R., Reviriego, P., & Larrabeiti, D. (2024). Supporting Dynamic Insertions in Xor and Binary Fuse Filters With the Integrated XOR/BIF-Bloom Filter. *IEEE Transactions on Network and Service Management*.

Moeini, M., & Shahhoseini, H. S. (2019). Parallel Rabin-Karp algorithm for string matching using GPU. In *10th International Conference on Information and Knowledge Technology (IKT 2019)*.

Moeini, M., & Shahhoseini, H. S. (2022, February). Improved Rabin-Karp Algorithm Using Bloom Filter. In *2022 27th International Computer Conference, Computer Society of Iran (CSICC)* (pp. 1-5). IEEE.

Putri, R. E., & Siahaan, A. (2017). Examination of document similarity using the Rabin-Karp algorithm. *International Journal of Recent Trends in Engineering & Research, 3*(8), 196-201.

Rahim, R., Ahmar, A. S., Ardyanti, A. P., & Nofriansyah, D. (2017, December). Visual Approach of Searching Process using Boyer-Moore Algorithm. In *Journal of Physics: Conference Series* (Vol. 930, No. 1, p. 012001). IOP Publishing.

Rahim, R., Zulkarnain, I., & Jaya, H. (2017, September). A review: search visualization with Knuth Morris Pratt algorithm. In *IOP Conference Series: Materials Science and Engineering* (Vol. 237, No. 1, p. 012026). IOP Publishing.

Hakak, S. I., Kamsin, A., Shivakumara, P., Gilkar, G. A., Khan W. Z. , & Imran, M. (2019). Exact String Matching Algorithms: Survey, Issues, and Future Research Directions. *IEEE Access*, *7*, 69614-69637. https://doi.org/10.1109/ACCESS.2019.2914071

Siahaan, Andysah Putera Utama. (2018). Rabin-Karp elaboration in comparing pattern based on hash data. *International Journal of Security and Its Applications*, *12*(2), 59-66.

Singh, A., Garg, S., Kaur, R., Batra, S., Kumar, N., & Zomaya, A. Y. (2020). Probabilistic data structures for big data analytics: A comprehensive review. *Knowledge-Based Systems, 188,* paper 104987.

Author's Biography

Beatrice A. Escoto is an incoming graduate with a bachelor's degree in computer science from Pamantasan ng Lungsod ng Maynila. She actively participates in tech-related extracurricular activities to gain exposure and enhance her skills. She is dedicated to developing her expertise in UI/UX Design and Quality Assurance, demonstrating strong leadership abilities. She interned at the university's ICT Office, where she contributed as a mobile developer and analyst. In her free time, Beatrice enjoys visiting museums and reading tech articles, finding peace and inspiration in these activities.

Acemon Jassene G. Ladines is an aspiring backend developer from Pamantasan ng Lungsod ng Maynila, focusing on creating RESTful APIs. During an internship at the university's ICT Office, he honed his technical skills and gained practical experience. He is dedicated to continuous learning, upskilling on his own, and staying updated with the latest industry trends. Outside of his technical pursuits, Acemon enjoys watching series and working out, finding balance and relaxation through these activities.

Abigail Grace G. Macahilig, a driven student at Pamantasan ng Lungsod ng Maynila pursuing a Bachelor of Science in Computer Science. She is known for her unrelenting devotion to academic success, as well as a strong desire for ongoing learning and improvement. Throughout her undergraduate career, she distinguished herself through her continuous mastery of UI/UX design and Quality Assurance, demonstrating not only technical expertise but also an in-depth understanding of the symbiotic relationship between technology and human experience. Her professional experience includes an IT internship at Solid Shipping Lines Corporation, where she honed her skills in quality assurance and project management. Macahilig's journey is a testament to her passion for both learning and technological innovation.

Professor Mark Anthony S. Mercado is an academic expert with information technology experience. After graduating from Arellano University with a Bachelor of Science in Computer Science with a Major in Information Technology and pursuing a Master's degree in Information Technology at the Technological University of the Philippines, he laid a solid foundation for his career. His dedication to lifelong learning and high standards in academia is apparent. His career in academia has been marked by a broad range of positions and responsibilities, including leadership positions as LMS Manager, Team Leader, and Subject Coordinator, as well as teaching positions at Arellano University and La Consolacion College Manila. Prof. Mercado's standing in the academic world was further cemented when he was appointed Chairperson of the Information Technology Department at Pamantasan ng Lungsod ng Maynila on August 17, 2023.

Dr. Dan Michael A. Cortez is the Vice President of Research, Academic, and Extension Services at the Pamantasan ng Lungsod ng Maynila. He previously held the position of Program Chairperson in the Computer Science Department. With ten years of teaching experience, he earned a Bachelor of Science in Information Technology and a Master of Science in Information and Communications Technology from the same institution. He earned his Doctorate Degree in Information Technology from the Technological Institute of the Philippines-Quezon City Campus. A member of PSITE-NCR and the Philippine Computing Society, as well as an author and researcher in information technology.

Raymund M. Dioses currently holds the positions of chairperson of the Computer Science Department and Assistant Professor I in the College of Information Systems and Technology Management. Before this, he spent eight years as a faculty member and chair of the Computer Education Department at CORE Gateway College Inc. and worked in the Senior High School Department of the Department of Education for five years. He holds a Master of Arts in Education, majoring in Educational Management from CORE Gateway College, and a Bachelor of Science in Computer Science from St. Jude College. At Nueva Ecija University of Science and Technology, he is also pursuing a Master of Information Technology, majoring in Computer Education. His vast background in academic administration and teaching has greatly improved his teaching abilities.

Jonathan C. Morano currently is a Lecturer I at Pamantasan ng Lungsod ng Maynila. He has more than 20 years of experience working and teaching in the field of Information Technology. Mr. Morano earned a Bachelor of Science in Computer Science from the Technological University of the Philippines. He acquired a Master of Arts in Teaching with a major in Information Technology from Central College of the Philippines, as well as a Master of Science in Information Technology from La Consolacion University Philippines. He is also the author of numerous research publications in information technology, both locally as well as internationally.

Jamillah S. Guialil is a 27-year-old resident of Intramuros, Manila. She graduated with a Bachelor of Science in Computer Studies, majoring in Computer Science, in 2018. Currently, she is pursuing a Master of Information Technology at Pamantasan ng Lungsod ng Maynila. She also teaches as a part-time faculty member at the College of Information Systems and Technology Management at the same university. Her devotion to the field of technology is evident in her pursuit of excellence in both her studies and teaching.